LISTEN.
THINK.
SOLVE.℠

INTEGRATED PRODUCTION & PERFORMANCE SUITE

Performance & Visibility

FactoryTalk® View Machine Edition



USER'S GUIDE VOLUME 2

PUBLICATION VIEWME-UM005E-EN-E–August 2007
Supersedes Publication VIEWME-UM005D-EN-E

ALLEN-BRADLEY • ROCKWELL SOFTWARE

Rockwell Automation

Contacting Rockwell

Customer Support Telephone — 1-440-646-3434

Online support — http://support.rockwellautomation.com

Copyright Notice

© 2007 Rockwell Automation Technologies Inc. All rights reserved. Printed in USA.

This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation Technologies Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation Technologies Inc. is strictly prohibited. Please refer to the license agreement for details.

Trademark Notices

Allen-Bradley, ControlLogix, FactoryTalk, PLC-2, PLC-3, PLC-5, Rockwell Automation, Rockwell Software, RSLinx, RSView, the Rockwell Software logo, and VersaView are registered trademarks of Rockwell Automation, Inc.

The following logos and products are trademarks of Rockwell Automation, Inc.:

RSMACC, RSLogix, Data Highway Plus, DH+, RSView, FactoryTalk View, RSView Studio, FactoryTalk View Studio, PanelBuilder, PanelView, RSView Machine Edition, RSView ME Station, and WINtelligent.

FactoryTalk Activation, FactoryTalk Administration Console, FactoryTalk Alarms and Events, FactoryTalk Automation Platform, FactoryTalk Services Platform, FactoryTalk Diagnostics, FactoryTalk Directory, FactoryTalk Live Data, RSAssetSecurity, and FactoryTalk Security.

PanelView, RSLinx Classic, RSLinx Enterprise, SLC 5, and SLC 500

Other Trademarks

ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows ME, Windows NT, Windows 2000, Windows Server 2003, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association Inc. (ODVA)

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control is a registered trademark of the OPC Foundation.

Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This product is warranted in accord with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at anytime without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Doc ID VIEWME-UM005E-EN-E
August 2007

18 Working with components

This chapter describes:

- which editors have components.
- working with components.
- printing information in components.

For information on working with particular editors, see the chapters later in this guide.

Editors that have components

With some editors you enter information in a single window or a tabbed dialog box. Other editors allow you to create multiple components, such as graphic displays or message files. Each component is stored in a separate file, with its own name.

You can create components in these editors:

- Graphics (components include graphic displays, global object displays, and graphic libraries, each in their own folder)
- Parameters
- Local Messages
- Information Messages
- Data Log Models
- Macros
- RecipePlus

The Explorer window lists the components you create under the icon for the editor you used to create the component.

You can use the Images editor to copy bitmap images into your application (but not to create new images). Each image you copy is listed as a component under the editor.

To view a list of components for an editor

1. Click the + symbol to the left of the editor icon, or double-click the editor name.

Working with components

This section describes how to perform basic operations that are common to all components.

For information on working with components in particular editors, see the chapters later in this guide.

Creating components

To create a new component, do one of the following

- Drag the component icon into the workspace.
- Right-click the editor, and then click New.
- On the File menu, select New, and then click the type of component to create.

Opening components

To open a component, do one of the following

- Double-click the component.
- Right-click the component, and then click Open.
- Drag the component from the Explorer window to the workspace.

Saving components

The Save tool is available when the active component contains unsaved changes.

To save a component



Save tool

1. On the File menu, click Save, or click the Save tool.



2. If this is the first time you're saving the component, type a name in the Component name box, and then click OK.



Close button on the title bar

Closing components

To close a component



Close button in component

1. On the File menu, click Close, or click the Close button on the component's title bar.

   Some components have a Close button that you can click to close the component.

Before the component closes you are prompted to save unsaved changes, if there are any.

Adding components into an application

You can use the same components in more than one application by adding components into an application.

For example, if you want to use the same graphic display in Application A and Application B, create the display in Application A, then add the graphic display component from Application A to Application B.

When you add the component into Application B, changes you make to the component in Application B are not reflected in the component in Application A.

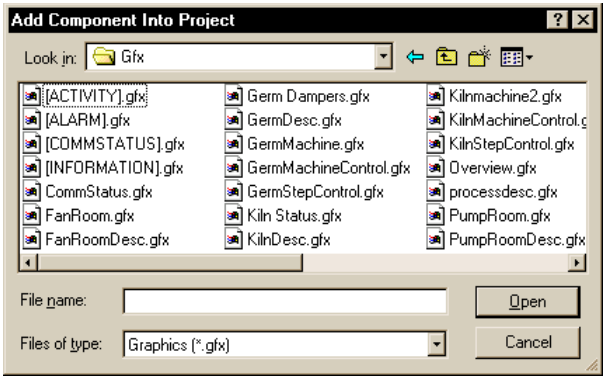
All the language strings associated with the component in Application A are copied into Application B. For example, if you add an information message file for which three sets of language strings have been defined, all three sets of strings are copied into Application B, regardless of which languages have been set up for Application B. For more information about using multiple languages, see Chapter 12.

To add a component from Application A into Application B

1. In Application B, right-click the type of editor that was used to create the desired component.

   For example, to add a graphic display component, right-click the Displays icon in the Graphics folder.

2. Click Add Component Into Application.



3. In the dialog box, navigate to the component to add (in Application A's folder), and then click the component's file name.

Click and Shift-click to select a group of components, or Ctrl-click to select multiple individual components.

For information about application folders and files, see Chapter 4, <italics>Working with applications.

4. Click Open. The components are listed under the editor in the Explorer window in Application B.

If you later modify the component in Application A, you can add the component into Application B again using the same steps as described above.

Using Add Component Into Application with graphic displays, graphic libraries, and global object displays

The Add Component Into Application menu command is also useful for:

- adding graphic displays to your application's Libraries folder.
- using libraries as graphic displays in your application.
- moving libraries into the Displays folder so you can export their strings for translation.
- creating global object displays.

For information about graphic libraries, see page 19-14. For information about global object displays, see page 25-7.

Deleting components

Deleting a component deletes it from the Explorer window and from the hard disk.

To delete a component

1. Right-click the component, and then click Delete.

Removing components

You can remove a component from the Explorer window but leave it on the hard disk, in case you want to use the component in another application.

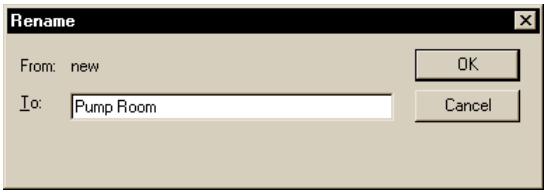
To remove a component

1. Right-click the component, and then click Remove.

Renaming components

To rename a component

1. Right-click the component, and then click Rename.



2. In the To box, type the new name.
3. Click OK.

Duplicating components

The Duplicate option is useful for creating multiple similar components. For example, you could create a graphic display to use as a template, then duplicate the display each time you want to use the template.

To duplicate a component

1. Right-click the component, and then click Duplicate.
2. In the Component name box, type a name for the duplicate component.
3. Click OK.

Printing

Each component has a Print item on its File menu.

To print a component's contents

1. Open the component.
2. On the File menu, click Print.
3. Click OK.

For information about selecting a printer and printing at run time, see page 2-13 in Volume 1 of the *FactoryTalk View Machine Edition User's Guide.*

19 Using graphic displays

This chapter describes the Graphics, Images, and Local Messages editors in the Graphics folder and outlines how to:

- use the Graphics editor.
- set up graphic displays.
- create a background for your displays.
- use graphic libraries.
- import graphic images.
- create local messages in your displays.
- print graphic displays at run time.

Before creating graphic displays, specify project settings. Project settings determine important aspects of your graphic display such as size and position. For more information, see page 4-11.

For information about global object displays and the Parameters editor, see Chapter 25.

About graphic displays and graphic objects

A graphic display represents the operator's view of plant activity. The display can show system or process data and provide operators with a way to write values to an external data source. The data source can be memory or a device such as a programmable controller or an OPC® server.

Operators can print the display at run time to create a visual record of tag values, object states, or current messages.

The elements that make up a graphic display are called graphic objects. You can create objects in the Graphics editor, or copy them from a global object display, from a graphic library, or from another application.

For information about creating and copying graphic objects, see Chapter 20.

You can use up to 1,000 tags per graphic display. This limit includes the tags contained in embedded variables.

Before you begin

Before you begin, plan your displays. Think about what information the operator needs to see, and the best way to provide the information. For example:

- Does the operator need to know the exact speed of a conveyor belt, or just whether the belt is moving, jammed, or stopped?
- Do different users need to have access to different types of information?
- Do you need to limit access to certain types of information?

Also consider the runtime environment and how the operator will use the application:

- Does the runtime computer have a touch screen, mouse, keyboard, or some combination of these?
- How will the operator navigate through the displays of the application?
- Will the application be available in multiple languages?

Review the chapters on planning, security, and navigation before you begin creating displays. Browse through the sample applications for design ideas. Map out a display hierarchy. Then create a graphic display to use as a template.

The time you spend planning your displays will make your application easy to use and will save you time in the long run.

For information about	See
Planning your displays and creating a template	Chapter 3
Setting up application security to control access to displays	Chapter 11
Setting up display navigation and creating a display hierarchy	Chapter 13
Setting up how objects are used at run time	Chapter 21

Using the Graphics editor

The Graphics editor opens when you create or open a graphic display, global object display, or graphic library. Each display is stored in the Displays folder. You can open and work on multiple graphic displays at the same time.

Creating and opening graphic displays

To create a graphic display

1. In the Graphics folder, do one of the following:
   - right-click Displays and then click New.
   - click the New Display tool.
   - drag and drop the Displays icon into the workspace.



New Display tool

2. On the Edit menu, click Display Settings to open the Display Settings dialog box and specify settings for the display.

   For more information, see page 19-11.

3. Create the objects you want to put in the display.

   For information about creating graphic objects, see Chapter 20.



Save tool

4. On the File menu click Save, or click the Save tool.
5. In the Component name box, type a name for the display, and then click OK.

   The display is added to the list in the Displays folder.

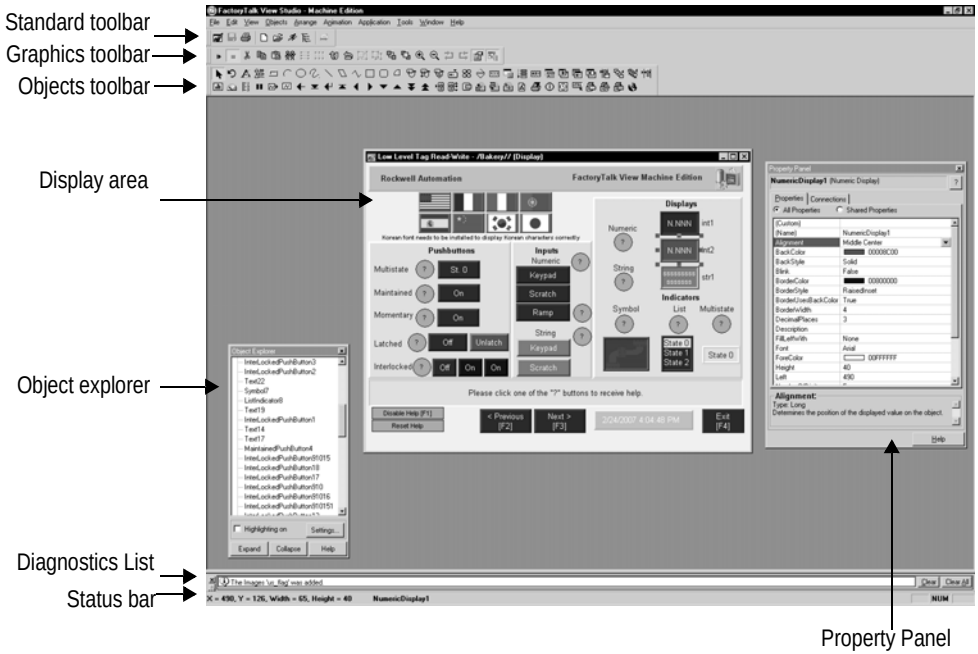
   The display is created as a Replace display by default, but you can change it to an On Top display in the Display Settings dialog box. For more information about display types, see page 19-12.

To open a graphic display

- In the Graphics folder, open the Displays folder and then double-click the display name, or right-click the display name and then click Open.

  You can also drag and drop the display from the Explorer window to an empty area in the FactoryTalk® View Studio workspace.

This example shows a graphic display with the Explorer window closed.



The Graphics editor has special items on the View and Edit menus, and extra toolbars.

For details about using the Graphics editor, see Help.

Importing and exporting graphic displays

Each graphic display's information is contained in a file called *Displayname*.gfx.

The Graphics Import Export Wizard in FactoryTalk View Studio allows you to export this information to an XML file, or to import a graphic display XML file.

You can export the display information, import the graphic display XML file to another FactoryTalk View application, and then modify it to suit your requirements. Or, you can modify the file before importing it.

You can use the Graphics Import Export Wizard to import graphic display information that has been created using an external programming tool or editor, or you can import a FactoryTalk View XML file.

For more information about importing and exporting graphic display files, see Appendix H.

Tools and tips for working in the Graphics editor

This section describes features of the Graphics editor that help you create your displays. It describes how to:

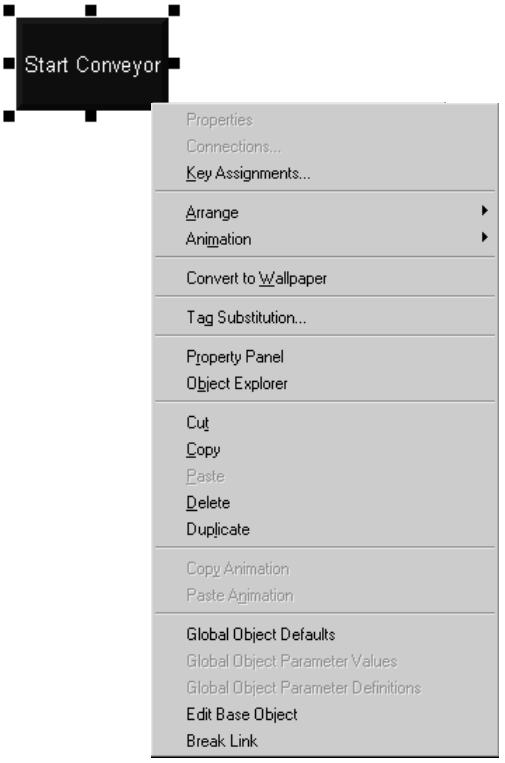
- use context menus and toolbars to perform actions quickly.
- view displays in grayscale to emulate the appearance of a PanelView™ Plus terminal.
- set up a grid that you can use to position and size objects precisely.
- zoom in when you need to look at details closely.
- correct mistakes.
- test your displays as you work.

Using context menus

No matter where you are in the Graphics editor, you can open a menu by clicking the right mouse button. This is often quicker than moving the mouse up to the menus at the top of the screen.

The items on the menu depend on the cursor's location. For example, when you right-click an object, the menu contains the most common actions you can perform on that object.

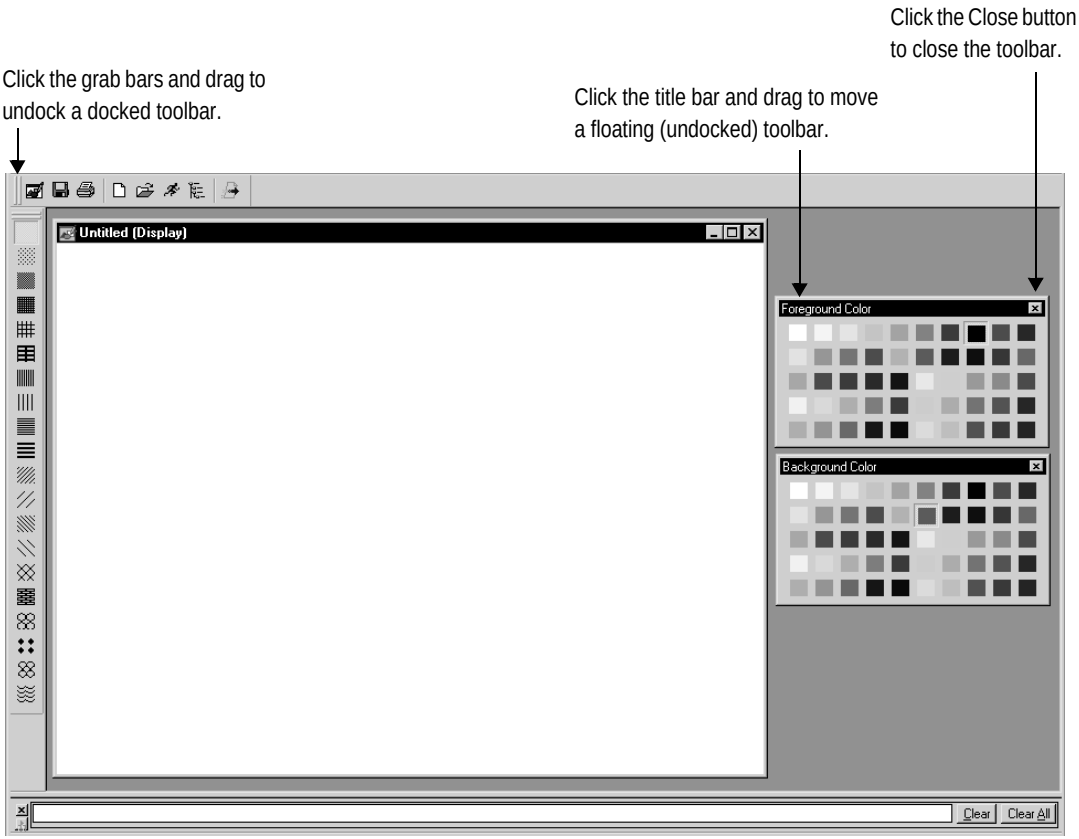
When you right-click an object...



...a menu opens. It contains menu items for working with the selected object.

Using the toolbars

The toolbars provide another convenient way to perform an action quickly. You can customize your workspace by dragging the toolbars to any location on the screen. You can also dock them to the inside edges of the FactoryTalk View Studio workspace.


Click the Close button to close the toolbar.
Click the grab bars and drag to undock a docked toolbar.
Click the title bar and drag to move a floating (undocked) toolbar.
Untitled (Display)
Foreground Color
Background Color
Clear
Clear All

To display a toolbar

- On the View menu, select Toolbars, and then click the toolbar to display.

The menu displays a check mark beside the toolbars that are already open.

To undock a toolbar

- Click the double “grab bars” at the left or top of the toolbar, and then drag. Press the Ctrl key to prevent accidental redocking.

To move an undocked toolbar

- Click the toolbar's title bar, and then drag. Press the Ctrl key to prevent accidental redocking.

To dock a toolbar

- Click the toolbar's title bar, and then drag to any edge of the workspace.

To close a toolbar

- On the View menu, select Toolbars, and then click the name of the toolbar to close, or click the toolbar's Close button.

Showing displays in grayscale

You can set displays to appear in grayscale, especially for developing applications that will run on the PanelView Plus grayscale terminals. When Show Displays in Grayscale is checked (on the View menu), all open displays will change from full color to grayscale. When Show Displays in Grayscale is cleared, all open displays will appear in color.

Show Displays in Grayscale does not work in Test Application mode.

Third party ActiveX® objects will not appear in grayscale if Show Displays in Grayscale is checked.

Using the grid

To size and position objects precisely, use the grid. You can change the grid settings at any time during the drawing process.

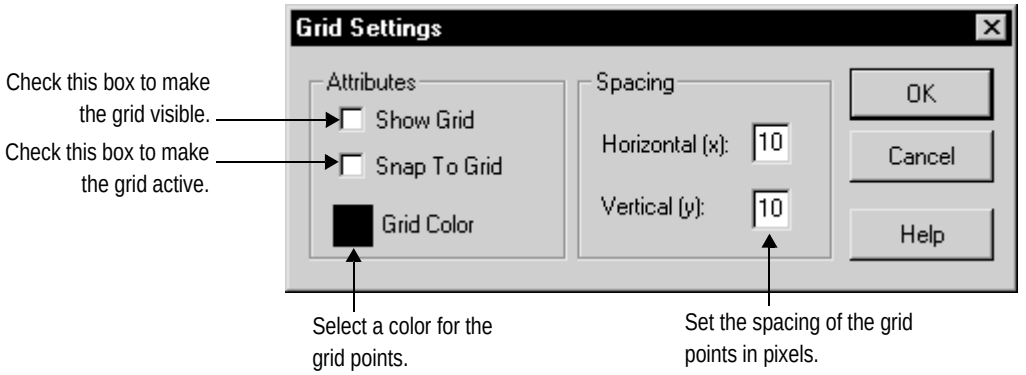
The grid can be active or passive. When the grid is active, all the objects you draw or position are pulled to the closest grid point. This makes it easy to align and size objects. When the grid is passive, it is visible but does not affect the position of your objects.

Make the grid passive to position an element between the grid lines. Make the grid active, and the next object you draw or place is automatically aligned with the grid. Making the grid active does not affect the placement of existing objects.

The grid is visible during application development only. It is not visible at run time.

To set up the grid

1. On the View menu, click Grid Settings, or right-click an empty area of the display and then click Grid Settings.



2. Specify the color and spacing of the grid points.
3. To turn on the grid, click Show Grid.

   When the Grid Settings dialog box is not open you can turn the grid on by clicking Show Grid on the View menu.
4. To make the grid active, click Snap To Grid.

   When the Grid Settings dialog box is not open you can make the grid active by clicking Snap On on the View menu.
5. Click OK.

To make the grid passive

- In the Grid Settings dialog box, clear the Snap To Grid check box, or on the View menu click Snap On to toggle the option off.

  There is no check mark beside the menu item when it is turned off.

To turn off the grid

- In the Grid Setting dialog box, clear the Show Grid check box, or on the View menu, click Show Grid to toggle the option off.

Zooming in and out

To magnify or reduce your view of a graphic display, use Zoom In and Zoom Out. Zoom In magnifies objects; Zoom Out reduces magnification.



Zoom In tool

To zoom in on objects

1. Select the objects you want to zoom in on.
2. On the View menu, click Zoom In, or click the Zoom In tool.



Zoom Out tool

To zoom out

- On the View menu, click Zoom Out, or click the Zoom Out tool.

Correcting mistakes

If you change your mind about something you did, you can undo the action. If you change your mind again, you can redo the action.

You can undo and redo all the operations you performed since you last saved the display, one operation at a time.

The operations you perform between opening and closing a dialog box are treated as a single operation. Operations you perform in the Property Panel are treated as separate operations.



Undo tool

To undo an operation

- On the Edit menu, click Undo, or click the Undo tool.

To redo an operation



Redo tool

- On the Edit menu, click Redo, or click the Redo tool.

Testing your displays as you work

To test the objects in your displays as you work, use the Test Display tool to switch to test mode. When you are finished testing, switch back to edit mode to continue editing.



Test mode is not the same as running the display. It does not change the appearance or position of the display as set up in the Display Settings dialog box. Alarm and information messages are not displayed, although if communications are set up, tag values are read and written. Display navigation, data logging, parameters, and macros do not work in this mode. If you want to test these features, test the application as described on page 14-2.

If you set up local messages and graphic objects to use multiple languages, the messages and objects are displayed in the current application language during test mode.

Using the Diagnostics List when in test mode

The Diagnostics List shows messages about system activities when you test your graphic displays. You can specify the types of messages to display in the Diagnostics List, move the list, resize it, and clear the messages in it.

For information about using the Diagnostics List, see page 2-5. For information about specifying the types of messages to show in the Diagnostics List, see page 10-4.



Test Display tool

To use test mode

- On the View menu, click Test Display, or click the Test Display tool.



Edit Display tool

To return to edit mode

- On the View menu, click Edit Display, or click the Edit Display tool.

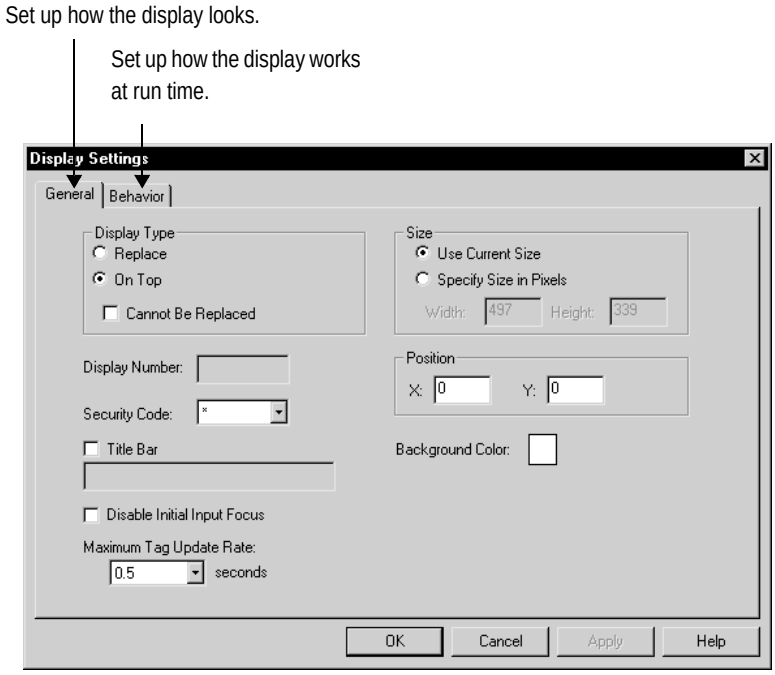
Setting up graphic displays

To set up a graphic display, specify its type, background color, and runtime behavior in the Display Settings dialog box. You can specify and edit the display settings at any time while you work on your display.

Specifying display settings

To specify display settings

- On the Edit menu, click Display Settings, or right-click an empty area of the display and then click Display Settings.



For details about the options in the Display Settings dialog box, see Help.

About display types

Replace displays

Replace is the default display type. Replace displays are full-sized displays. They use the project window size specified in the Project Settings editor.

For more information about the project window size, see page 4-11.

At run time, you can have only one Replace display open at a time. When the operator opens a Replace display, this is what happens:

- The Replace display that was open closes.
- On Top displays that do not use the Cannot Be Replaced option are closed.
- The new Replace display opens.
- On Top displays that use the Cannot Be Replaced option remain open, on top of the new Replace display.

If the operator attempts to open a Replace display that is already open (for example, using a goto display button to which the same display is assigned), the display does not close and FactoryTalk View sends an error message to FactoryTalk® Diagnostics.

On Top displays

Use the On Top option to create “pop-up” displays that open on top of the current Replace display. Usually, On Top displays are smaller than Replace displays, so the operator doesn’t lose track of display navigation.

You can open multiple On Top displays. If more than one On Top display is open, the display that has focus, or had the most recent focus, appears on top.

When an On Top display closes, the display that had the most recent focus appears on top.

Use the Cannot Be Replaced option if you want the On Top display to remain open when a new Replace display is opened.

On Top displays do not have a Close button in the title bar. Be sure to create a close button graphic object in On Top displays so the operator can close them.

The operator cannot move an On Top display by dragging its title bar. The runtime position of the display is fixed (according to the position settings defined for the display).

You can specify unique titles for On Top displays. You can use embedded variables in the title, and the title text can switch languages at run time.

Resizing displays

The project window size is used for all Replace displays. If you change the project window size after you have designed any graphic displays, you have the option of scaling graphic displays.

If you choose to scale graphic displays, all Replace and On Top displays are resized and the objects in them are scaled to fit the new size. You can also specify whether to scale the font size and border size of the graphic objects, and the size of graphic images in the Images folder.

If you choose not to scale displays when you change the project window size, Replace displays are resized, but any objects in the displays remain the same size and in the same position as before. On Top displays are not resized.

For information about specifying the project window size, see page 4-11.

Changing the display area while working

You can change the display area of a display while you’re working on it by dragging the border of the display area. For example, you might want to make the display area smaller so you can see parts of two displays in order to drag and drop objects between them.

Changing the display area in this manner does not change the display size for Replace displays. For On Top displays that are sized with the "Use Current Size" option, dragging the border does resize the display.

Another way to arrange multiple displays while working on them is to use the options on the Windows menu. For example, Tile Horizontal arranges all the open displays with as much of the top part of each display showing as possible.

Creating a background for your display

You can create a background for your graphic display by converting graphic objects to wallpaper. When objects are converted to wallpaper, they are locked into position and become an unchanging background for the other objects in the display.

Converting objects that do not need to be animated or updated with tag values can significantly improve the runtime performance of a graphic display.

Objects that have been converted to wallpaper cannot be selected or edited until you unlock the wallpaper. Also, animations attached to the wallpaper objects are not in effect. However, animations are restored when you unlock the wallpaper.

Similarly, any tags or expressions assigned to an object become inactive when the object is converted to wallpaper. Connections are restored if you unlock the wallpaper.

To manage a number of objects easily, group the objects, and then convert the group to wallpaper.

To convert objects to wallpaper

1. Select the objects to convert.

   For information about selecting objects, see page 20-23.

2. On the Edit menu, select Wallpaper, and then click Convert to Wallpaper.

   For a single object, you can right-click it and then click Convert to Wallpaper.

To unlock the wallpaper

1. On the Edit menu, select Wallpaper, and then click Unlock All Wallpaper.

   All objects in the graphic display are converted back to their original state.

Using graphic libraries

FactoryTalk View Studio comes with a set of libraries, contained in the Libraries folder. As with a public library, the graphics libraries can provide you with source materials and reference information.

Each graphic library consists of a graphic display, with the file extension .gfx. The library displays contain graphic objects that you might find useful in your own application. Many of the objects are preconfigured with animation. For example, see the Conveyor parts library.

You can:

- look at the objects and displays to get ideas for your own application.
- drag and drop (or copy and paste) objects from the libraries into your own displays.

  For information about copying objects from a library into your graphic display, see page 20-42.
- use the objects as they are or change them to suit your needs.
- create your own libraries of objects.
- create libraries of displays that are translated into multiple languages.
- use libraries as displays in your application.

To open a graphic library

1. In the Explorer window, open the Graphics folder, and then open the Libraries folder.
2. Double-click the library name, or right-click it and then click Open.

Creating graphic libraries

You can create a graphic library in the Libraries folder, or create a graphic display in the Displays folder and then add the display to the Libraries folder.

To create a graphic library

1. In the Graphics folder, right-click Libraries and then click New.
2. Create the objects you want to put in the library.

   For information about creating graphic objects, see Chapter 20.



Save tool

3. On the File menu click Save, or click the Save tool.
4. In the Component name box, type a name for the library, and then click OK.

   The library is added to the list in the Libraries folder.

You can also create a graphic display and then use Add Component Into Application to add the display to the Libraries folder.

To add a graphic display to the library

1. Create the graphic display.
2. In the Explorer window, in the Graphics folder, right-click the Libraries icon.
3. Click Add Component Into Application.
4. In the dialog box, navigate to the Gfx folder, and then click the .gfx file for the display to use.

   The Gfx folder is located in:

   \Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects*Application Name* (Windows® 2000)

   or

   \Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME\HMI projects*Application Name* (Windows XP or Windows Server 2003 R2)

   where *Application Name* is the name of your application.
5. Click Open. The display is copied into the Libraries folder.

Using libraries as displays in your application

The graphic libraries are available on the development computer, but do not appear at run time. To use a library as a graphic display at run time, you must add the library into your application's folder of graphic displays.

If the library doesn't contain strings for languages supported by the current application, the undefined strings are displayed with question marks (?).

To use a library as a display in your application

1. In the Explorer window, in the Graphics folder, right-click the Displays icon.
2. Click Add Component Into Application.
3. In the dialog box, navigate to the Libraries folder, and then click the .gfx file for the library to use.

   The Libraries folder is located in:

   \Documents and Settings\All Users\Documents\RSView Enterprise\ME (Windows 2000)

   or

   \Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME (Windows XP or Windows Server 2003 R2)

4. Click Open. The library is copied into the Displays folder.

Using libraries to store displays with multiple languages

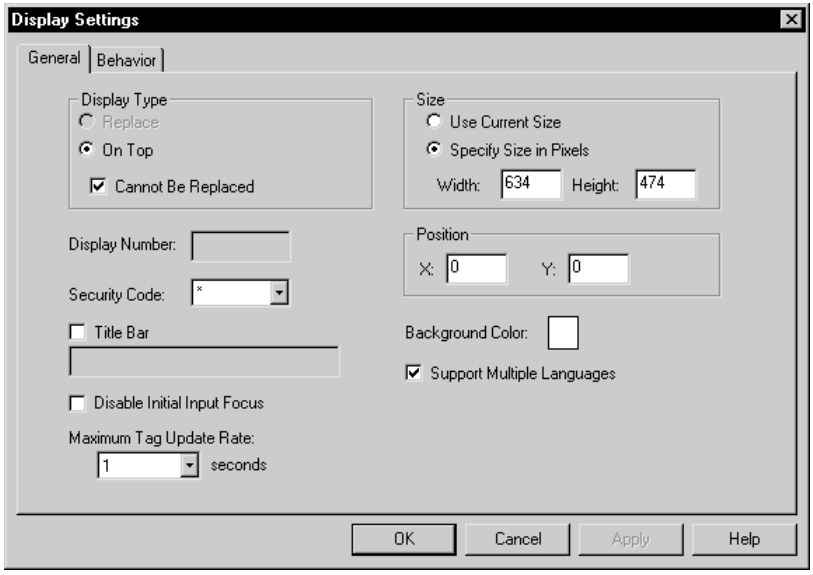
The maximum number of languages an application can use is 40 during development, and 20 at run time. Since the libraries are stored outside of the HMI project folder and are available to all applications, they do not have this limit. However, since they are not stored in the HMI project folder, you cannot export the strings in library displays for translation. Instead, export a graphic display for translation, import the translated strings, then add the display into the library.

When you use a library display in your application, make sure you add to your application the languages that you plan to use from the library.

To save a library display with multiple languages

1. Create a graphic display.
2. Export the text for your application, which will include the text used in the graphic display.
3. Translate the text strings into each desired language, saving the file with a new name for each language.
4. Import the files for all the new languages.
5. Add the graphic display to the Libraries folder.

   The library is created with the option Support Multiple Languages selected (in the Display Settings dialog box).



For detailed information about importing and exporting to use multiple languages, see Chapter 12.

To save a library display with more than 40 languages

1. Create a display in one application, with up to 40 languages, and add it to the library, as described in the previous section.
2. Add the display into a new application, with up to 40 different languages in the new application.
3. Export the text for your application, which will include the text used in the graphic display.
4. Translate the text strings into each desired language, saving the file with a new name for each language.
5. Import the files for the new languages.
6. Add the graphic display (with the same name as the display in step 1) to the Libraries folder.

The new languages are added to the library. Make sure the display contains the same objects as the original display. Otherwise the new display will overwrite the previous display, and all the original language strings will be undefined.

What is displayed

When you open a library that supports multiple languages, the strings are displayed using the current application language, if available. If the library does not contain the current language, the strings are displayed as single question marks. Similarly, if the library contains the language but not all the strings are defined for the language, undefined strings are displayed as single question marks.

For more information about using multiple languages with graphic libraries, see Chapter 12.

Location of library components

The Libraries folder is located in:

\Documents and Settings\All Users\Documents\RSView Enterprise\ME (Windows 2000)

or

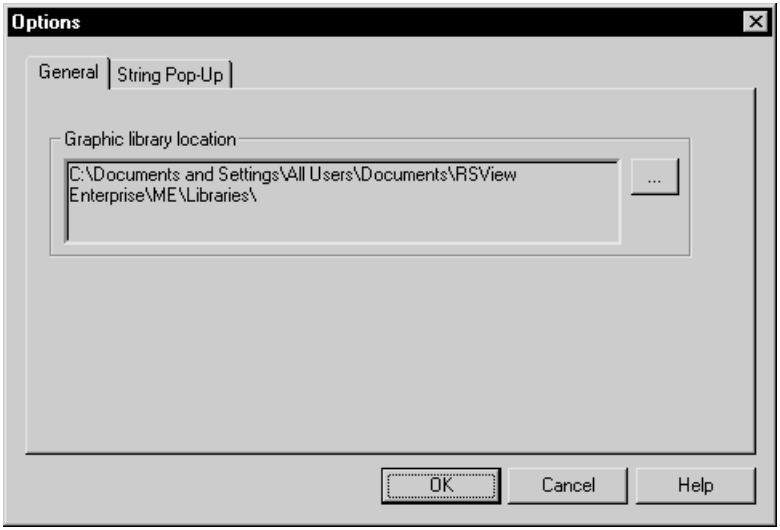
\Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME (Windows XP or Windows Server 2003 R2)

If desired, you can specify a different directory in which to store library components. If you specify a different directory, FactoryTalk View saves graphic libraries that you add or create in the new directory. Similarly, when you open a library component FactoryTalk View looks for the component in the specified directory.

If you specify a different directory but want to use the libraries that come with FactoryTalk View, use My Computer or Windows Explorer to copy the library component files into the directory you've specified, or else change the path back to the default path when you want to open a FactoryTalk View library component.

To specify a new path for graphic libraries

1. On the Tools menu, click Options.



For details about using the Options dialog box, see Help.

Importing images into your application

You can use these types of external graphic files in your graphic displays:

- .bmp—bitmap images
- .jpg—JPEG images
- .dxf—AutoCAD® files
- .wmf—Windows metafiles

You import and place .dxf and .wmf files in your displays in one step. Once imported, they are converted to drawing objects. For more information about using these types of files in your displays, see page 20-20.

Bitmap images that come with FactoryTalk View Studio

FactoryTalk View Studio comes with sets of bitmaps that are useful for illustrating graphic objects and displays:

- arrows
- DIN symbols
- ISA symbols

- keyboard button symbols such as Enter and Page Up
- parts such as buttons, conveyors, pipes, tanks, and valves

The symbols and most of the arrows are monochrome (that is, use only two colors, one for the foreground and one for the background).

You can set up the foreground and background colors for the monochrome images that you use for your objects.

For color images, including JPEG images, you can specify whether to use a transparent or solid background. If you select the transparent background style, the black portions of the image become transparent.

For detailed information about setting up objects, see Help.

Location of bitmap files

The bitmap files that come with FactoryTalk View Studio are stored in this directory:

\Documents and Settings\All Users\Documents\RSView Enterprise\Images (Windows 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView Enterprise\Images (Windows XP or Windows Server 2003 R2)

The images are in folders called Arrows, DIN, ISA, ListKey, and Parts.

Importing bitmap and JPEG images

To use bitmap and JPEG images, you can:

- import the images first, and then place them in your displays as needed. This method is useful for images that you use to illustrate your displays.

  For information about placing images in your display once you've imported them, see page 20-15.

- use the Image Browser to import images as needed while setting up your graphic objects. This method is useful for images that you use as labels on your graphic objects.

  For information about using the Image Browser, see page 19-23.

- copy and paste images from the graphic libraries.

  For information about copying and pasting objects, see page 20-42.

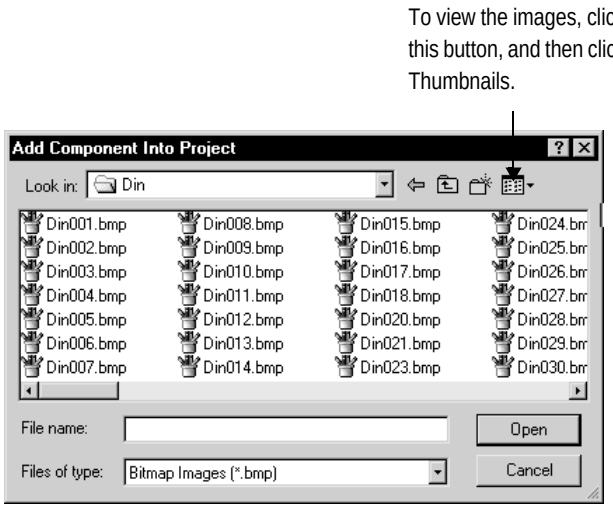
- copy and paste images from one application to another.

For information about opening two applications at once, see page 4-9.

If you will be using images that have more than 256 colors, before importing the images set up your video adapter to display more than 256 colors. This will ensure that the colors of imported images appear the same as in the original. For more tips about using images, see page 19-24.

To import bitmap and JPEG images

1. In the Graphics folder, right-click Images and then click Add Component Into Application.

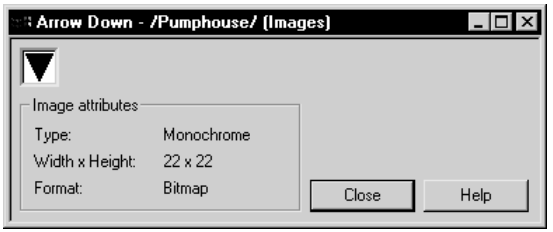


2. In the Files of type box, select the type of image to add.
3. Navigate to the directory where the .bmp or .jpg file is stored, and then click the file name.

   Shift-click or Ctrl-click to select additional files.
4. Click Open to add the selected files to the list in the Images folder.

To view an image that you've already imported

1. In the Images folder, double-click the image name, or right-click it and then click Open.



Using the Image Browser to import images

Use the Image Browser to import images as needed while you set up graphic objects.

In the Image Browser you can:

- import images into the application.
- select the image to use on a graphic object.
- delete images from the application.

To open the Image Browser, use one of these methods



Browse button

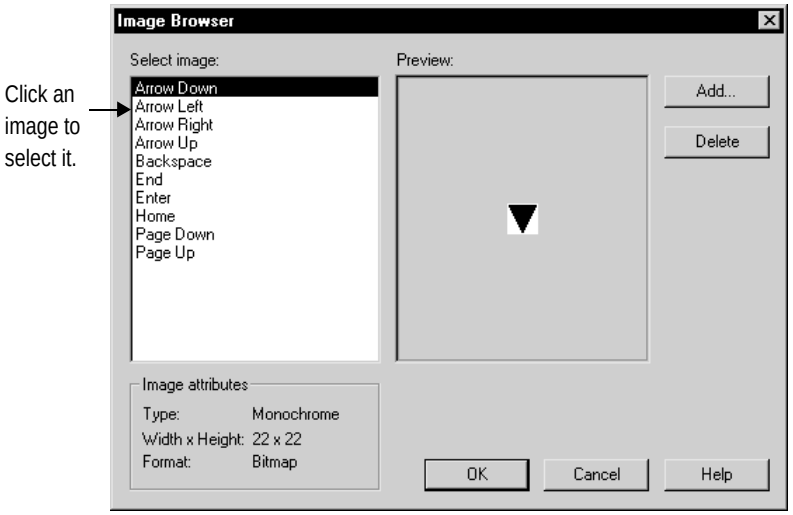
- In an object's Properties dialog box, click the Browse button next to the Image box.

  Depending on the type of object, the Image box could be located on the General tab, Label tab, or States tab.

  For information about opening the Properties dialog box, see page 20-28.

- With one or more objects selected, in the Property Panel click the Image property, and then click the Browse button.

  For information about opening the Property Panel, see page 20-29.



For details about using the Image Browser to import, select, and delete images, see Help.

You can also remove an image by clicking it in the Images folder and then right-clicking Remove or Delete. For more information, see page 18-4.

Tips for using images

Using bitmaps versus JPEG images

When deciding whether to use a bitmap image or a JPEG image, consider these points:

- FactoryTalk View supports 256 color (grayscale) and 16 million color JPEG images only. For all other color types, use bitmaps.
- For large color images (16 million color, 320 x 240 pixels or larger), JPEG images load faster than bitmaps. In all other cases, bitmaps load faster than JPEGs.
- Color JPEG images have a much smaller file size than the equivalent bitmap image, and therefore require less disk space at run time.

Guidelines for using images

Images consume Windows resources, so when using graphic images use the lowest color depth possible.

The more colors you use, the more memory is consumed, and the longer the image takes to load and display.

This bitmap type	Consumes this many bits per pixel
Monochrome	1
16 color	4
256 color	8 (1 byte)
64 K color	16 (2 bytes)
16 million color	24 (3 bytes)

For example, a 24-bit bitmap image that measures 800x600 pixels consumes 1440 KB of memory. If the bitmap color depth is decreased to 256 colors, the image might have minor color loss, but the new image uses only 480 KB of memory.

Use images with a similar pixel size to the size of the FactoryTalk View object on which the image will be placed. FactoryTalk View resizes the image to fit the object, but if you use an image that is much larger than the object, the display will be slow to open at run time, due to the time required to resize the image.

Using local messages

Use local messages to give the operator ongoing information about the status of devices and processes. For example, you might use local messages to describe the status of a device whose condition cannot be shown graphically with complete accuracy.

The messages you create in the Local Messages editor are displayed in local message display objects in graphic displays. You can use multiple local message display objects in your application, and link each object to a different file of messages. Or, you can use the same file of messages for multiple local message display objects.

Local messages versus information messages

Use local messages to give the operator information in a specific graphic display while the display is open. To give the operator information no matter which display is open, use information messages.

For details about information messages, see Chapter 27.

Summary of steps

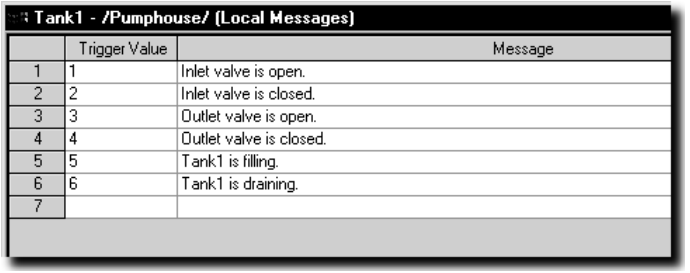
Follow these steps to set up local messages:

1. In the Local Messages editor, set up the messages and their trigger values.
2. In the Graphics editor, create local message display objects in the graphic displays in which you want the messages to appear at run time. For each local message display, assign a tag or expression to the Value connection and specify the file of messages to display.

   For information about creating graphic objects, see Chapter 20. For information about setting up local message display objects, see page 21-53.

Using the Local Messages editor

Use the Local Messages editor to create one or more files of local messages. Each file is stored in the editor's folder. You can open and work on multiple message files at the same time.


Tank1 - /Pumphouse/ (Local Messages)

	Trigger Value	Message
1	1	Inlet valve is open.
2	2	Inlet valve is closed.
3	3	Outlet valve is open.
4	4	Outlet valve is closed.
5	5	Tank1 is filling.
6	6	Tank1 is draining.
7		

You can define up to 10,000 messages in each message file.

For details about using the Local Messages editor, see Help.

Preparing to set up local messages

As your application is running, information is continually sent to the data source about the state of various processes. For example, your application might be monitoring whether a valve is open or closed, or the temperature in a boiler. Values representing the status of these processes are sent to the data source.

The data source

The FactoryTalk View documentation uses the term data source as a generic term that includes all possible sources of tag data, for both data server tags and HMI tags. The data source can be memory or a device such as a programmable controller or OPC server. FactoryTalk View writes values to and reads values from the data source. The data source

is set up to exchange information (in the form of numeric or string values) between FactoryTalk View and the machine that your application is controlling.

Identifying the tags and values to monitor

To set up local messages, determine which tags associated with machine processes to monitor, and identify the values for those tags that will trigger local messages.

For information about creating HMI tags, see Chapter 7.

How local messages work

These are the key parts of the local message system:

- Local message files—text files containing lists of messages, with a numeric trigger value for each message
- Local message display object—a graphic object that displays local messages when the Value connection assigned to the object matches a message's trigger value
- Value connection—a tag or expression. When the value of this connection matches a message's trigger value, the local message display object displays the associated message.

The local message display object always appears in the graphic display it's placed in, whether or not there is a message to display. However, the operator does not see the message unless the object is located in the display the operator is currently viewing.

The following example shows how the key parts of the local message system work together.

Example: Displaying local messages

This example shows how to notify the operator of the status of a hoist.

1. Create a tag called Hoist_Status. This tag points to an address in a programmable controller that is linked to sensors on the hoist. The tag has five possible values:

The tag has this value	When the hoist has this status
1	At bottom
2	Raising
3	Stopped between the top and bottom
4	Lowering
5	At top

2. In the Local Messages editor, create these messages with trigger values matching the values that will be sent to the Hoist_Status tag:

Trigger value	Message
1	The hoist is ready to rise
2	The hoist is raising the pallet.
3	The hoist has stopped.
4	The hoist is lowering the pallet.
5	The hoist is finished rising.

   Save the message file with the name "Hoist status."

3. In the Graphics editor, create a local message display object. In the object's Properties dialog box, assign the Hoist_Status tag to the Value connection, and select the Hoist status message file.

At run time, when the operator views the graphic display containing the local message display object, the status of the hoist is displayed.

Local messages and trigger values

Create messages associated with each tag value that you want to inform the operator about. Assign each message a trigger value, and set up the data source to send the trigger value to the Value connection. You can use both HMI and data server tags

The trigger value can be any non-zero integer value (positive or negative). Trigger values do not need to be contiguous, but they must be unique for each message. For example, you could use trigger values of 1, 2, and 3, or values of 10, 20, and 30.

If you use an analog tag or an expression, you can use any non-zero integer or floating point value to trigger an alarm. Floating point values are rounded to the nearest integer. For information about how values are rounded, see page 7-2.

Trigger values cannot be 0. Digital tags have two possible values, 0 and 1. Therefore, if you use a digital tag you can only use the value 1 to trigger a message. If you want to use a digital tag to trigger two different messages, create an expression that adds 1 to the digital tag's value. That way, you can use the trigger values 1 and 2.

When the Value connection's value is 0, the local message display object is cleared.

Creating local messages in multiple languages

FactoryTalk View 5.00 supports local messages in multiple languages. When you create local messages, they are in the current application language. You can export the local messages for translation and then import them back into the application. For details, see Chapter 12.

Language switching local messages in RSView ME Station 4.00

For applications that will run in RSView ME Station version 4.00, use the CurrentLanguage( ) expression function to specify message offsets in the local message file. In the file, divide your messages into sections for each language. For information about the CurrentLanguage( ) function, see page 23-15.

How the local message display graphic object works

When you open a graphic display at run time, FactoryTalk View reads the value of the Value connection and updates the local message display object based on the value.

What is displayed

- If the Value connection is unassigned, the display is filled with question marks (?).
- The Value connection is rounded to the nearest integer. If the value does not match any of the trigger values in the specified message file, the display is filled with question marks.

  For information about how values are rounded, see page 7-2.
- If the message is too long to fit in the object, the last displayed character is replaced with an asterisk (*).
- When the Value connection's value is 0, the display is cleared.
- If you set up local messages in multiple languages, messages are displayed in the current application language. When a language switch occurs, the message that was already in the local message display remains in the language that it originally appeared in. New messages are displayed in the new language.

Printing displays

You can print your graphic displays on the development computer. This might be useful if you want other people to review the displays before implementing the application, or if you want to keep a visual record of the displays.

You can also print graphic displays at run time, to provide a printed record of process values such as trend data.

For information about printing on the development computer, see page 2-13.

Printing displays at run time

When you print a display at run time, everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

For information about specifying which printer to use at run time for applications that will run on a personal computer, see page 15-10.

For information about specifying printer options for applications that will run on a PanelView™ Plus or PanelView Plus CE terminal, see the *PanelView Plus Terminals User Manual*.

To print graphic displays at run time, use one or both of these methods

- Create display print buttons in the graphic displays you want to print. At run time, the operator presses the buttons to print the displays.

  For information about creating graphic objects, see Chapter 20.

- Assign a tag or expression to the Remote Display Print connection (in the Global Connections editor). When the value of the tag or expression changes from 0 to a non-zero value, the current displays are automatically printed.

  Program the data source to trigger the change as often as you want the data printed.

  For more information about setting up remote display printing, see page 8-5.

20 Using graphic objects

This chapter describes the types of graphic objects and outlines how to:

- create graphic objects, including drawing and ActiveX® objects.
- select and deselect objects and use the Object Explorer.
- use the Properties dialog box and Property Panel to set up objects.
- color and name objects.
- test how objects look in different states.
- assign tags and expressions to objects.
- replace tags using tag substitution.
- use tag placeholders.
- move, copy, duplicate, resize, reshape, and delete objects.
- group and ungroup objects, and edit group objects.
- arrange objects and lock objects into position.

For information about setting up graphic objects once you've created them, see Chapter 21. For information about setting up global objects, see Chapter 25.

For examples of how to set up objects, see the sample applications that come with FactoryTalk® View Studio. The Help also provides examples of how to use objects.

Types of graphic objects

The elements that make up a graphic display are called graphic objects. Use objects to control your process, machines, and application.

FactoryTalk View comes with a complete range of configurable objects such as push buttons, list selectors, bar graphs, and trends. Some objects interact with the data source, allowing the operator to change or view tag values. For example, the operator can push a button to set a tag value to 1, causing a programmable controller to start a conveyor belt. Other objects are used to control your application. For example, there are button objects that you can use to change displays and scroll through lists.

FactoryTalk View also comes with drawing objects that you can use to illustrate your graphic displays. The drawing objects include text, bitmap images, and geometric and freehand shapes.

FactoryTalk View also supports the use of ActiveX objects—third-party software components that you can use to control processes and display information. The ActiveX objects available depend on which third-party applications are installed on your development computer. For example, products such as Microsoft® Visual Basic®, Rockwell Automation® RSTools®, and Microsoft Office provide ActiveX objects that are configurable in FactoryTalk View Studio.

Using the tables

The tables beginning on the next page will help you choose which objects to use to control your application, machines, and process. The tables group the objects according to function, provide an overview of what each object does, and list cross-references to more detailed information about using the objects.

About connections

Many of the objects use connections to interact with the data source. A connection is the link between the object and the data source. Depending on an object's function, the object may have more than one connection. For example, a momentary push button has a Value connection and an Indicator connection. The Value connection is used to set a value at the data source, and the Indicator connection is used to display the data source value in your graphic display.

By assigning tags or expressions to an object's connections, you control the flow of data between the application and the data source, which in turn controls your process or machines. Assign tags or expressions to an object's connections in the Connections tab of the object's Properties dialog box (see page 20-26) or in the Connections tab of the Property Panel (see page 20-29).

Illustrating your displays

Use this graphic object	To do this
Text	Create text for labels or instructions in the display. See page 20-13.
Image	Place images in your display. For more information, see page 20-14.
Panel	Draw rectangles that have borders. See page 20-16.
Arc	Draw an arc (a segment of an ellipse or circle's perimeter). See page 20-16.
Ellipse	Draw ellipses and circles. See page 20-17.
Freehand	Draw freehand shapes as you would with a pen on paper. See page 20-17.
Line	Draw straight diagonal, horizontal, and vertical lines. See page 20-18.
Polygon	Draw a series of connected straight lines forming a closed shape. See page 20-18.

Use this graphic object	To do this
Polyline	Draw a series of connected straight lines. See page 20-18.
Rectangle	Draw rectangles and squares. See page 20-19.
Rounded rectangle	Draw rectangles and squares with rounded corners. See page 20-20.
Wedge	Draw a filled segment of an ellipse or circle. See page 20-16.

Controlling the application

Use this graphic object	To do this
Goto display button	Open a graphic display. For details, see page 13-4.
Return to display button	Close a display and return to the previous display. For details, see page 13-5.
Close display button	Close a display. Can also send a value to a tag. For details, see page 13-6.
Display list selector	Select a display to open from a list of displays. For details, see page 13-6.
Display print button	Print the current display. For details, see page 19-30.
Language switch button	Switch the application language. For details, see page 21-16.
Login button	Open the Login dialog box and then log in. For details, see page 11-12.
Logout button	Log out of the application. For details, see page 11-13.
Password button	Change the current user's password. For details, see page 17-4.
Shutdown button	Stop the application and shut down FactoryTalk View ME Station. For details, see page 13-7.
Goto configure mode button	Stop the application and open the FactoryTalk View ME Station dialog box. For details, see page 13-4.
Print alarm history button	Print a report of alarm messages in the alarm log file. You can print a report for all alarms, or for a specified alarm trigger. For details, see page 21-56.
Print alarm status button	Print a report of the status of alarms, including how many times each alarm was triggered and the time in alarm. You can print a report for all alarms, or for a specified alarm trigger. For details, see page 21-57.

Starting and controlling processes

Use this graphic object	To do this
Momentary push button	Start a process or action by sending one value to the tag when pressed, and another value when released. For details, see page 21-20.
Maintained push button	Toggle between two values by sending one value to the tag when pressed, and a second value the next time the button is pressed and released. This button is useful for changing a setting within a machine or process, but not for starting the machine or process. For details, see page 21-21.
Latched push button	Start a machine or process. The button remains set (latched) until the process is complete. For example, use this button to start a bag filling machine. When the process is complete (the bag is full), the button is reset (unlatched) by the Handshake connection. For details, see page 21-23.

Use this graphic object	To do this
Multistate push button	Cycle through a series of values. Each time the operator presses the button, the value for the next state is sent to the tag. When the button is in its last state, pressing it changes the button to its first state and writes out the first state value. This button is useful when you want the operator to see and select multiple options in sequence, using a single button. The button displays the current state of an operation by showing a different color, caption, or image to reflect the different states. For details, see page 21-24.
Interlocked push button	Use a group of buttons to send values to the same tag. When the operator presses one button in the group, the button's value is sent to the tag, and the button remains highlighted as long as the tag value is the same as the button's value. Pressing another button in the group releases the first button, and sends a new value to the tag. For details, see page 21-26. You can also use a single interlocked push button to send a value to a tag.
Ramp button	Increase or decrease the value of a tag by a specified integer or floating-point value. For example, use two ramp buttons together to create a raise/lower control. For details, see page 21-27.
Control list selector	Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator can scroll through the list to select a different state. The value assigned to the selected state is written to the tag. For details, see page 21-47.
Piloted control list selector	Select from a list of states for a process or operation. The list is highlighted to show the current state, and the operator or a remote device such as a programmable controller can scroll through the list to select a different state. For details, see page 21-50.
Drawing object with horizontal or vertical slider animation	Set the value of a tag by dragging the slider object. The pixel position of the slider is translated into a value that is written to the tag. If the value of the tag is changed at the data source, the position of the slider changes to reflect this. For information about animation, see Chapter 22.
ActiveX object	Change tag values using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. When the object's property value changes, the new value is written to the associated tag. For details, see page 20-21.

Displaying processes and values graphically

Use this graphic object	To display this
Bar graph	Numeric values in bar graph format. The bar graph increases or decreases in size to show the changing value. For details, see page 21-45.
Gauge	Numeric values in dial format. The gauge's needle moves around the dial to show the changing value. For details, see page 21-46.
Scale	A static indication of the range of values for a bar graph. For details, see page 21-47.
Multistate indicator	The state of a process, on a panel that changes its color, image, or caption to indicate the current state. Each state is set up to correspond to a numeric tag value. For details, see page 21-39.
Symbol	The state of a process, using a monochrome image that changes color to indicate the current state. Each state corresponds to a numeric tag value. This object is useful for showing the state of a process or operation at a glance. For details, see page 21-40.
List indicator	The state of a process, using a list of possible states with the current state highlighted. Each state is represented by a caption in the list, and corresponds to a numeric tag value. This indicator is useful if you want to view the current state, but also want to see the other possible states. For sequential processes, the list can alert the operator about what happens next in the process. For details, see page 21-42.
Trend	Historical or current numeric tag values, plotted against time or displayed in an XY plot (where one or more tags' values are plotted against another tag's values to show the relationship between them). For details, see Chapter 28.
Time and date display	Display the current time and date. For details, see page 21-55.
ActiveX object	Data using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. The format of the data displayed depends on the object. For details, see page 20-21.
Drawing object with rotation, width, height, fill, color, or horizontal or vertical position animation	The value of a tag using a pictorial representation of the current value in relation to a range of possible values. For example, use rotation animation to show the tag value as a needle's position on a dial. For color animation, assign different colors to represent different values. For information about animation, see Chapter 22.

Working with lists, trends, alarm banners, and numeric input objects

Use this button	With this graphic object	To do this
Pause button	Trend	Toggle a trend between pausing and automatic scrolling.
Next pen button	Trend	Change the vertical axis labels for a trend to the scale for the next pen.
Backspace button	Control list selector Piloted control list selector	Move the cursor back to the highlighted item in the list.
End button	Lists and trends	List—move to the bottom item in the list. Trend—resume trend scrolling and move to the current (latest) data in the trend.
Enter button	Lists Alarm list and alarm banner	Select the item the cursor is pointing to. Acknowledge the currently selected alarm.
Home button	Lists and trends	List—move to the top item in the list. Trend—pause the trend and move to the earliest data in the trend.
Move left / right buttons	Trend	Pause the trend and scroll to the left or right.
Move up / down buttons	Lists, trends, and numeric input objects	List—move up or down one item in the list. Trend—scroll up or down to display higher or lower values on the vertical scale. Numeric input cursor point and numeric input enable button—ramp the value up or down.
Page up / down buttons	Lists	Move up or down one page in the list.
Acknowledge alarm button	Alarm list Alarm banner	Acknowledge and silence the selected alarm.
Alarm status mode button	Alarm status list	Change the type of alarms displayed in the alarm status list, from all alarms to active alarms to past alarms.
Clear alarm banner button	Alarm banner	Clear the alarm in the alarm banner without removing the alarm from the alarm log file and alarm lists.
Diagnostics clear button	Diagnostics list	Remove the selected message from all diagnostics lists.
Information acknowledge button	Information message display	Acknowledge the current message in the display.

For more information about using the buttons with lists and trends, see the topics on these pages:

For information about	See
Linking a button to a specific list, alarm banner, or trend object	Page 9
Using buttons with alarm lists, alarm banners, and alarm status lists	Page 35
Using buttons with information message displays	Page 6
Using buttons with diagnostics lists	Page 12
Using buttons with trends	Page 11

Unlike the buttons in the previous table, the following buttons do not work with specific graphic objects. You can use them to work directly with alarms in the alarm history and with the application's diagnostics messages.

Use this button	To do this
Acknowledge all alarms button	Acknowledge and silence all currently unacknowledged alarms, or the alarms for a specific alarm trigger.
Clear alarm history button	Remove alarms from the alarm log file and all alarm lists. You can remove all alarms, or just the alarms for a specific alarm trigger. You can also reset the cleared alarms. This resets the number of times an alarm has been triggered to 0, and the accumulated time in alarm to 0.
Reset alarm status button	Reset the number of times an alarm has been triggered to 0, and the accumulated time in alarm to 0, for all alarms.
Silence alarms button	Silence the audio indicator for all current alarms (on personal computers only).
Sort alarms button	Toggle between sorting alarms in alarm lists and the alarm log file by time and by trigger value.
Diagnostics clear all button	Remove all diagnostics messages from all diagnostic lists.

Entering and displaying numeric and string values

Use this graphic object	To do this
Numeric input enable button	Enter a numeric value and then write the value to a tag or an expression, or ramp a value at the data source. For details, see page 17-6.
Numeric input cursor point	Enter a numeric value and then write the value to a tag or an expression, or ramp a value at the data source. For details, see page 17-6.
String input enable button	Enter a string value and then write the value to a tag. For details, see page 17-10.
Numeric display	Display numeric tag values. For example, display the current temperature of an oven. For details, see page 21-28.
String display	Display string tag values. For example, set up the data source to generate strings that report on the state of a process of operation, or that provide the operator with instructions about what to do next. For details, see page 21-33.
RecipePlus button	Display data set and tag values for ingredients in the RecipePlus table. Write tag values from the selected data set to the data source. Write tag values from the data source to the selected data set or to a new data set. Save data set values from the table to a recipe file. Rename or delete recipe units. For details, see page 29-3.
RecipePlus selector	Select the recipe unit to display, download from, write to, rename, or delete. For details, see page 29-2.
RecipePlus table	Display and compare recipe data set values and tag values. Edit data set values. For details, see page 29-2.
ActiveX object	Enter or display data using a third-party object connected to an analog, digital, or string tag, including both HMI and data server tags. The format of the data entered or displayed depends on the object. For details, see page 20-21.

Displaying alarms and messages

Use this graphic object	To display this
Alarm list	Multiple alarm messages, including the time the alarms are triggered and acknowledged. For details, see page 9-32.
Alarm banner	A single unacknowledged alarm message. For details, see page 9-33.
Alarm status list	The status of alarms, including how many times an alarm has been triggered and how long it has been active. For details, see page 9-34.
Diagnostics list	Messages about system activity such as tag reads, tag writes, and communications errors. For details, see page 10-11.
Information message display	Messages about the process, prompts or instructions, and information about current states. For details, see page 27-6.
Local message display	Ongoing information about the status of devices or processes. For details, see page 19-29.

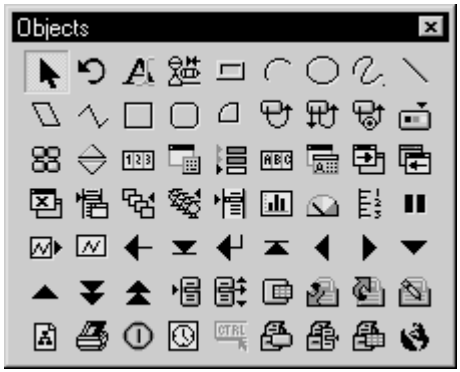
Selecting tools for creating graphic objects

The Objects menu in the Graphics editor contains items for creating objects, as well as items for selecting and rotating objects. You can also create most objects using the tools on the Objects toolbar.

Objects menu



Objects toolbar



Before you can create an object, you must select the object's tool, either by clicking a menu item or by clicking the tool on the toolbar. When you position a cursor over a tool on the toolbar, the name of the tool is displayed in a tooltip and in the status bar.

To select a tool

1. Click the tool on the toolbar or on the Objects menu. When you click a tool, the pointer changes to show which tool is active.



Freehand tool



Text tool

To deselect a tool, do one of the following

- Double-click an empty area in the display.
- Click the Select tool.
- Click another tool.



Select tool

For some drawing objects, double-clicking an empty area of the display creates another instance of the object. For these objects, to finish drawing, click the Select tool. For more information, see the instructions for creating drawing objects, beginning on page 20-12.

Before you begin creating objects

To size and position objects precisely as you create them, use the grid.

To use the grid

1. On the View menu, click Show Grid, and then click Snap On.

   For information about setting up the grid, see page 19-8.

The next sections describe how to create graphic objects. For information about setting up graphic objects once you've created them, see Chapter 21. For information about attaching animation to the objects you create, see Chapter 22.

Creating graphic objects

This section applies to graphic objects in general.

ActiveX objects and some drawing objects require extra steps to create them. For details about creating drawing objects, see the next section. For information about ActiveX objects, see page 20-21.

To create a graphic object

1. Select the tool for the object to create.
2. Click the mouse where you want to position the object, and then drag to draw a rectangle the general size you want the object to be.
3. Double-click the object to open its Properties dialog box.
4. In the dialog box, specify how the object looks, its behavior, and connections. For more information about the Properties dialog box, see page 20-26.

   You can also use the Property Panel to set up objects. For information about using the Property Panel, see page 20-29.

Creating drawing objects

The following instructions for creating drawing objects describe how to create the objects and then open the objects' Properties dialog boxes to set up how the objects look. For information about using the Properties dialog box, see page 20-26.

You can also use the Property Panel to set up objects. For information about using the Property Panel, see page 20-29.

Creating text

Choosing fonts

You can select any font you have installed, but TrueType™ and OpenType® fonts are recommended. These fonts can be resized easily, without losing text quality.

For PanelView™ Plus and PanelView™ Plus CE terminals, you must use TrueType fonts.

If you run an application on a computer that does not have the fonts you used when setting up the application, Windows® substitutes with the fonts that most closely match the fonts you specified.

Choosing fonts for language switching

If you are going to use the application with multiple languages, we recommend using Microsoft Sans Serif or Tahoma. These fonts allow for font linking to support the character sets of other languages. PanelView Plus and PanelView Plus CE terminals are shipped with font linking turned on.

For more information about font linking, see Help. For more information about setting up languages for your application, see Chapter 12.

Using the Size to fit option

If you use the Size to fit option, the size of the text object will likely change for different languages. Make sure you check all displays containing translated language strings.

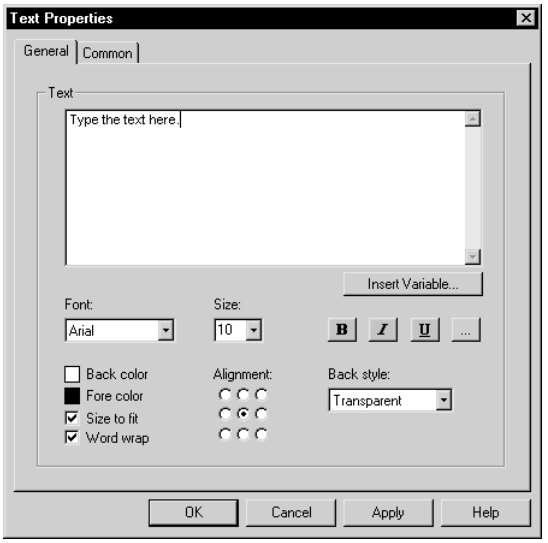
To create a text object



Text tool

1. Select the Text tool.
2. Click the mouse where you want to position the text, and then drag to draw a rectangle.

The Text Properties dialog box opens.



3. Select text options.

   For information about the options in the dialog box, see Help.

4. To close the dialog box, click OK.

   The text is positioned where you drew the rectangle.

5. To create another text object, move to a new area in the display, and then repeat steps 2 through 4.

After you've set up one text object to look the way you want, copy and paste it to create additional text objects with the same formatting. Then edit the text of the new objects.

To edit a text object

1. Double-click the text.

   The Text Properties dialog box opens.

2. In the Text box, click where you want to make the change. To delete text characters, use the Backspace and Delete keys.

Creating images

Use the image graphic object to place bitmap and JPEG images in your graphic displays.

Images support visibility animation only.

Using bitmaps and JPEG images

Before you can place a bitmap or JPEG image in a display, you must import the image into your application. For more information, see page 19-21.

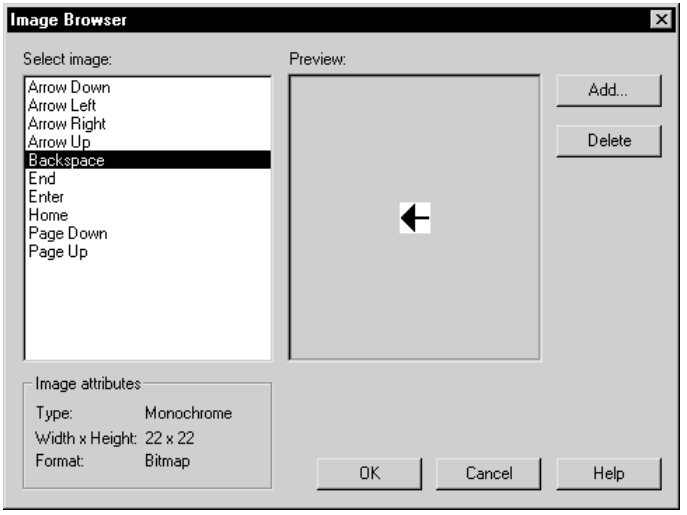
To place a bitmap or JPEG image in a display



Image tool

1. Select the Image tool.
2. Click the mouse where you want to position the image, and then drag to draw a rectangle.

   The Image Browser opens.



3. In the Select image list, click the image to place, and then click OK.

   For more information about using the Image Browser, see page 19-23.

   The image is placed where you drew the rectangle, but the actual size of the image is used, rather than the size of the rectangle you drew.
4. To change the image's attributes, double-click the image to open the Image Properties dialog box.
5. Select image options.

   For information about the options in the dialog box, see Help.
6. Click OK to close the dialog box.

Creating panels

Panels support visibility animation only.

To create a panel



Panel tool

1. Select the Panel tool.
2. Click the mouse where you want to position the panel, and then drag to draw a rectangle the general size you want the panel to be.
3. To change how the panel looks, double-click the panel to open the Panel Properties dialog box.
4. Select panel options.

   For information about the options in the dialog box, see Help.
5. To close the dialog box, click OK.

Creating arcs and wedges

Arcs and wedges are drawn in two steps: first you create an ellipse or circle, and then you reshape it into the segment you want.



Arc Hollow wedge Filled wedge

To create an arc or wedge

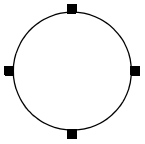


Arc tool

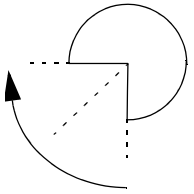
1. Select the Arc or Wedge tool.
2. Click the mouse where you want to position the object, and then drag to draw an ellipse or circle.

   To base the arc or wedge on a circle rather than an ellipse, hold down Ctrl while you drag.

   When you release the mouse button, a set of handles appears so you can decrease the angle of the wedge or arc from 360 degrees to the desired angle.



3. Click a handle, and drag the mouse to ‘cut out’ part of the circle.



4. To finish drawing, click the object.
5. To change how the object looks, click Properties on the context menu, or double-click the object to open its Properties dialog box.
6. Select arc or wedge options.

   For information about the options in the dialog box, see Help.
7. To close the dialog box, click OK.

You can also use the Arc and Wedge tools to reshape arcs and wedges. For more information, see page 20-45.

Creating ellipses and circles

Use the Ellipse tool to draw an ellipse or circle.

To create an ellipse or circle



Ellipse tool

1. Select the Ellipse tool.
2. Click the mouse where you want to position the object, and then drag to draw an ellipse or circle of the desired size.

   To draw a circle, hold down Ctrl while you drag.
3. To change how the object looks, double-click it to open the object’s Properties dialog box.
4. Select circle or ellipse options.

   For information about the options in the dialog box, see Help.
5. To close the dialog box, click OK.

Creating freehand shapes

Using the Freehand tool is similar to drawing with a pen on paper.

To create a freehand shape

1. Select the Freehand tool.



Freehand tool

2. Click and drag to create the shape you want.
3. To change how the object looks, double-click it to open the Freehand Properties dialog box.
4. Select freehand options.

   For information about the options in the dialog box, see Help.
5. To close the dialog box, click OK.

Creating lines

To create a line



Line tool

1. Select the Line tool.
2. Click where you want the line to start, and then drag from the beginning point to the end point.

   To draw horizontal or vertical lines, hold down Ctrl while you drag the mouse.
3. To change how the line looks, right-click the line and then click Properties on the context menu. The Line Properties dialog box opens.
4. Select line options.

   For information about the options in the dialog box, see Help.
5. To close the dialog box, click OK.

You can use the Polyline tool to convert the line into a polyline. For more information, see page 20-45.

Creating polygons and polylines

A polyline is a series of connected line segments. A polygon is a multi-sided object (with three or more sides). For example, use the polygon shape if you want to create triangles.



To create a polygon or polyline

1. Select the Polygon or Polyline tool.



Polygon tool



Polyline tool

2. Click and drag to create the first segment of the object. Release the mouse button.

   To draw horizontal or vertical lines, hold down Ctrl while you drag.



3. Move the mouse to where you want the next segment to end, and then click.

   Repeat this step until you have completed the object.



4. To finish drawing, double-click an empty area of the display, or click the Select tool.
5. To change how the object looks, double-click it to open the object's Properties dialog box.
6. Select polygon or polyline options.

   For information about the options in the dialog box, see Help.
7. To close the dialog box, click OK.

You can use the Polygon tool to reshape lines, polygons, polylines, and rectangles. For more information, see page 20-45.

Creating rectangles and squares

To create a rectangle or square



Rectangle tool

1. Select the Rectangle tool.
2. Click the mouse where you want to position the object, and then drag until the rectangle or square is the size you want.

   To draw a square, hold down Ctrl while you drag.
3. To change how the object looks, double-click it to open the object's Properties dialog box.
4. Select rectangle or square options.

   For information about the options in the dialog box, see Help.

5. To close the dialog box, click OK.

You can use the Polygon tool to reshape the rectangle into a polygon. For more information, see page 20-45.

Creating rounded rectangles and squares

Due to a Windows limitation, you cannot rotate rounded rectangles and rounded squares.

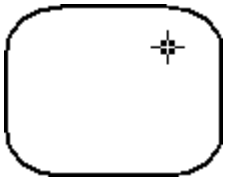
To create a rounded rectangle or square



Rounded rectangle tool

1. Select the Rounded Rectangle tool.
2. Click the mouse where you want to position the object, and then drag until the rectangle or square is the size you want.

   To draw a square, hold down Ctrl while you drag.
3. You can change how rounded the corners are by using the handle that appears inside the rounded rectangle. Click the handle and drag inward to increase roundedness, or outward to decrease roundedness.





4. To finish drawing, click the object.
5. To change how the object looks, click Properties on the context menu, or double-click the object to open its Properties dialog box.
6. Select rounded rectangle or square options.

   For information about the options in the dialog box, see Help.
7. To close the dialog box, click OK.

For information about using the Rounded Rectangle tool to reshape the rounded rectangle, see page 20-46.

Using .wmf and .dxf files

Windows metafiles (.wmf) and AutoCAD® (.dxf) files are converted to drawing objects (such as lines, ellipses, and polygons) when you import them. You can edit the drawing objects the same way you edit drawing objects that you create in FactoryTalk View.

Depending on the complexity of the metafile or AutoCAD file, the converted image could consist of 500 or more drawing objects. This would lead to long display load times. In this

case, it would be better to covert the .wmf or .dxf file to a bitmap, and then display the bitmap in an image object.

To place a .wmf or .dxf file in a display

1. On the Objects menu, click Import.
2. Click the mouse where you want to position the file, and then drag to draw a rectangle.
3. In the “Files of type” box, select the type of file to import.
4. Navigate to the directory where the file is stored, and then click the file to import.
5. Click Open.

   The file is converted to drawing objects and grouped, and then the grouped object is placed in the graphic display.

Using ActiveX objects

ActiveX objects use tags or expressions to exchange information with the data source.

The properties and connections available for a particular ActiveX object depend on the third-party vendor’s implementation.

ActiveX objects support visibility animation only.

If your application will run on a PanelView Plus CE terminal, the ActiveX object must be available for both the development system (running on Windows 2000, Windows XP, or Windows Server 2003 R2) and the PanelView Plus CE terminal platform. Make sure you install and register the Windows CE version of the ActiveX object on the PanelView Plus CE.

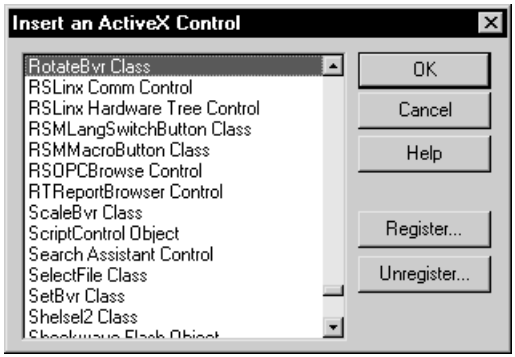
PanelView Plus terminals do not support ActiveX objects.

To create an ActiveX object



ActiveX Control tool

1. Select the ActiveX Control tool.
2. Click the mouse where you want to position the object, and then drag until the object is the size you want.



3. In the dialog box that opens, select the object to add to your graphic display.

   For information about the options in the dialog box, see Help.

4. Click OK.

   The object is placed in the display. Depending on how the third party implemented the object, it might be a different size than the rectangle you drew.

5. To specify the object's properties and assign tags or expressions to its connections, do one of the following:

   - Right-click the object, and then click Properties to open the object's Properties dialog box.

     Depending on how the third party implemented the object, it might not have a Properties dialog box. If the Properties menu item is not available, use the next method.

   - Right-click the object, and then click Property Panel.

6. In the Properties dialog box or Properties tab of the Property Panel, specify the object's properties.

7. If desired, set up the properties in the Common tab, as described on page 21-1.

8. In the Connections tab, assign tags or expressions to the object's connections.

9. To close the Properties dialog box, click OK. To close the Property Panel, click the Close button.

For information about using the Properties dialog box, see page 20-26. For information about using the Property Panel, see page 20-29.

Tools and tips for working with objects

This section describes features of the Graphics editor that help you work with the objects you create. It describes how to:

- select and deselect objects.
- use the Object Explorer to view and select objects.
- use an object's Properties dialog box to set up the object's properties and assign tags and expressions to its connections.
- use the Property Panel to set up individual and group object properties, and to assign tags and expressions to individual objects' connections.
- color objects.
- name objects.
- test how objects look in different states.

Selecting and deselecting objects

To work with an object, you must first select it. You can use the Select tool or the Object Explorer to select objects.

For information about using the Object Explorer, see page 20-24.

To select the Select tool



Select tool

- On the Objects menu, click Select, or on the Objects toolbar click the Select tool.

  The mouse pointer changes to a single arrow.

Selecting objects

To select	Do this
An object or group of objects	Click the object or group. In the Object Explorer, click the object or group.
An object within a group of objects	Double-click the group, and then click the object. In the Object Explorer, open the group, and then click the object.
Several objects	Click the first object, and then Ctrl-click additional objects.
All objects in an area	Click and drag diagonally to draw a selection border around the objects. Ctrl-click objects outside the border to add them to the selection.
All objects in the drawing area or in a group you are editing	On the Edit menu, click Select All, or press Ctrl+A.

Deselecting objects

To deselect	Do this
An object	Ctrl-click the object.
Several objects	Press and hold Ctrl, and then drag a selection border around the objects.
All selected objects	Click in the Drawing area, away from any objects.

Using the Object Explorer

Use the Object Explorer to view and select objects from a tree-list of all the objects in a display. Groups are listed as expandable items in the tree, with a + icon.

Use the Object Explorer to select an object that is hidden behind other objects in the display, without bringing the object to the front. Objects are listed in front-to-back order. The object you created most recently is at the front, unless you move it back using the Send to Back option. For more information about layering objects, see page 20-49.

You can also use the Object Explorer to highlight objects by object type, highlight objects that have animation attached, and to highlight objects with specific tag or expression assignments. For more information, see page 20-25.

The Object Explorer does not show wallpaper objects, nor objects within ActiveX composite objects.



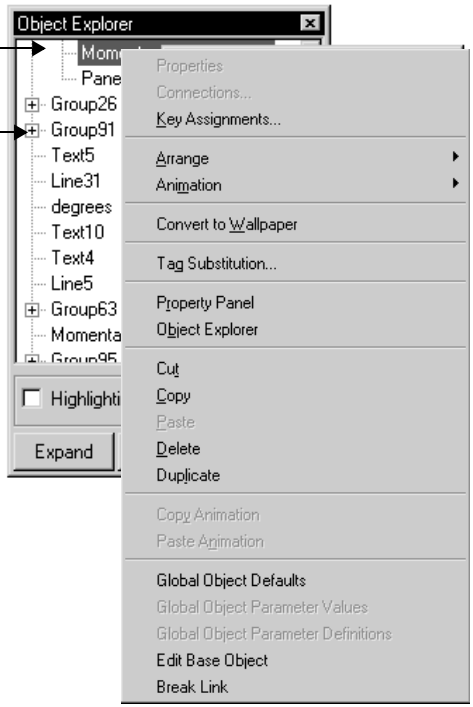
Show/Hide Object Explorer tool

To open the Object Explorer, use one of these methods

- On the Graphics toolbar, click the Object Explorer tool.
- On the View menu, click Object Explorer.
- Right-click an object, and then click Object Explorer.

Right-click an object to open its context menu.

Click the + icon to view the objects and groups within a group.



You can keep the Object Explorer open as you work in the Graphics editor. For more information about using the Object Explorer, see Help.

Highlighting objects in the Object Explorer

You can use the Object Explorer to highlight:

- specific types of objects
- objects that have animation attached

- objects that have a specific tag or expression assigned to them

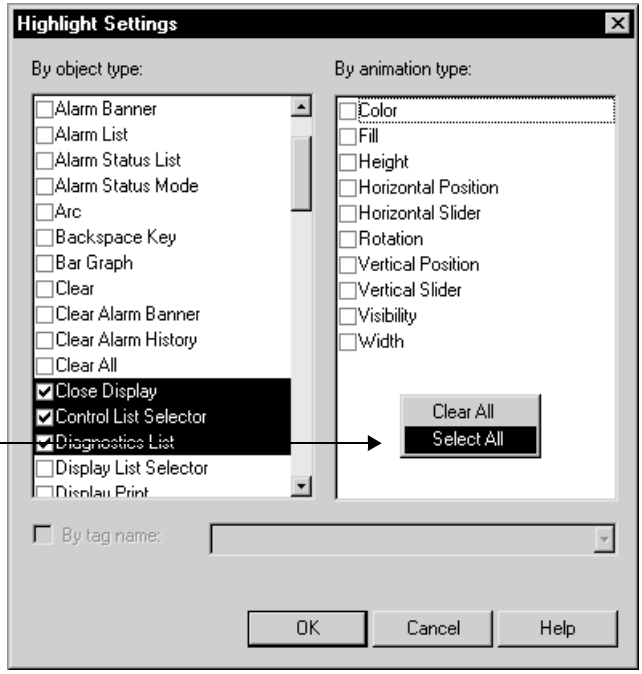
The objects are highlighted in red in the Object Explorer and in the graphic display. If your graphic display uses a red background, the highlighting is not visible in the graphic display.

To highlight objects in the Object Explorer

1. In the Object Explorer, click Highlighting on, and then click Settings.

To clear all the check boxes, right-click the list and then click Clear All.

To select all the check boxes, right-click the list and then click Select All.



For details about the options in the Highlight Settings dialog box, see Help.

Using the Properties dialog box

Every graphic object has a Properties dialog box that you can use to set up the object. Depending on how the vendor implemented the object, third-party ActiveX objects might have a Properties dialog box as well.

The Properties dialog box contains tabs that you can use to set up the object's properties and connections:

In this tab	Do this
General	Set up the object's appearance, audio indicator, and touch margins (for buttons), and settings that are unique to the object, such as the button action for a push button, whether to use key navigation to select the object, or whether to link a button to a specific object. For information about touch margins, see page 21-4. For information about key navigation, see page 21-8. For information about linking buttons to objects, see page 21-9.
States	Set up the states for the object, including the value for each state and whether to display a caption or image for the state. For information about checking that the states are set up the way you intended, see page 20-35.
Label	For objects that don't have multiple states, specify whether to use a caption or image on the object. For information about using the Image Browser to select an image to use in the label, see page 19-23.
Timing	Set up the object's auto repeat (see page 21-12) or Enter key handshaking (see page 21-13) settings.
Common	Set up the object's spatial properties, name and visibility. For details, see page 21-1.
Connections	Assign tags and expressions to the object's connections. For information about assigning tags and expressions, see page 20-36.

The tabs that are available depend on the object:

- Some objects have only General and Common tabs.
- If an object can have more than one state, the object's Properties dialog box contains a States tab.
- Some objects have unique tabs that are not listed in the table above.
- The tabs that an ActiveX object has depends on the vendor's implementation, though if you can assign tags or expressions to the object it has a Connections tab.

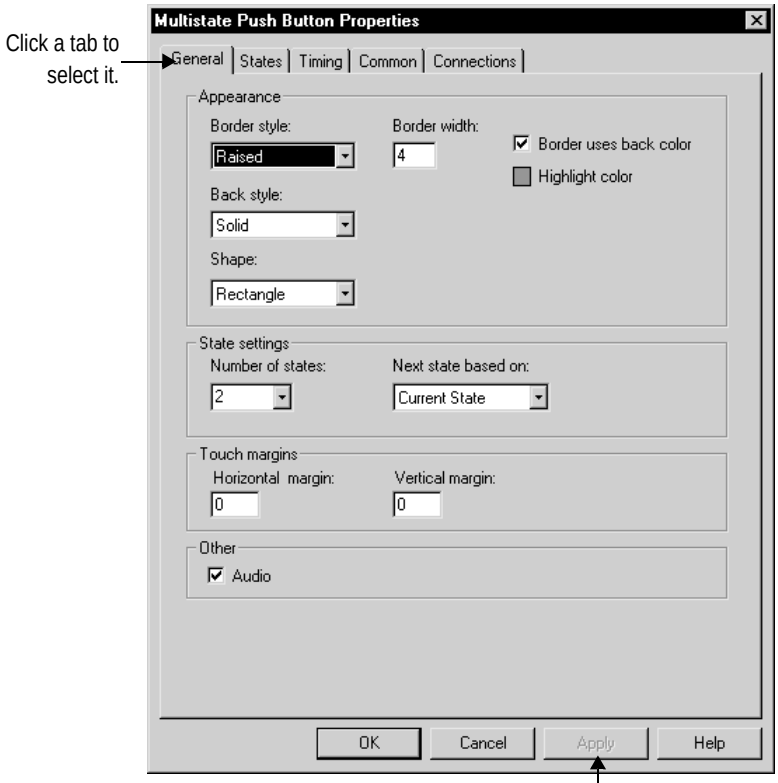
To open an object's Properties dialog box, use one of these methods

- Double-click the object.
- Right-click the object, and then click Properties.
- Select the object, and then on the Edit menu, click Properties.
- In the Property Panel, with the object selected, click the (Custom) property and then click the Browse button.



Browse button

- In the Property Panel, with the object selected, double-click the (Custom) property.
- In the Object Explorer, double-click the object.



You can also use the Property Panel to set up an object's properties, as described next.

For details about setting up a particular object, see Help.

Using the Property Panel

Use the Property Panel to modify the properties of graphic objects and assign tags and expressions to the objects.

The Property Panel is especially useful for making changes to the properties of multiple objects at the same time.



Show/Hide Property Panel tool

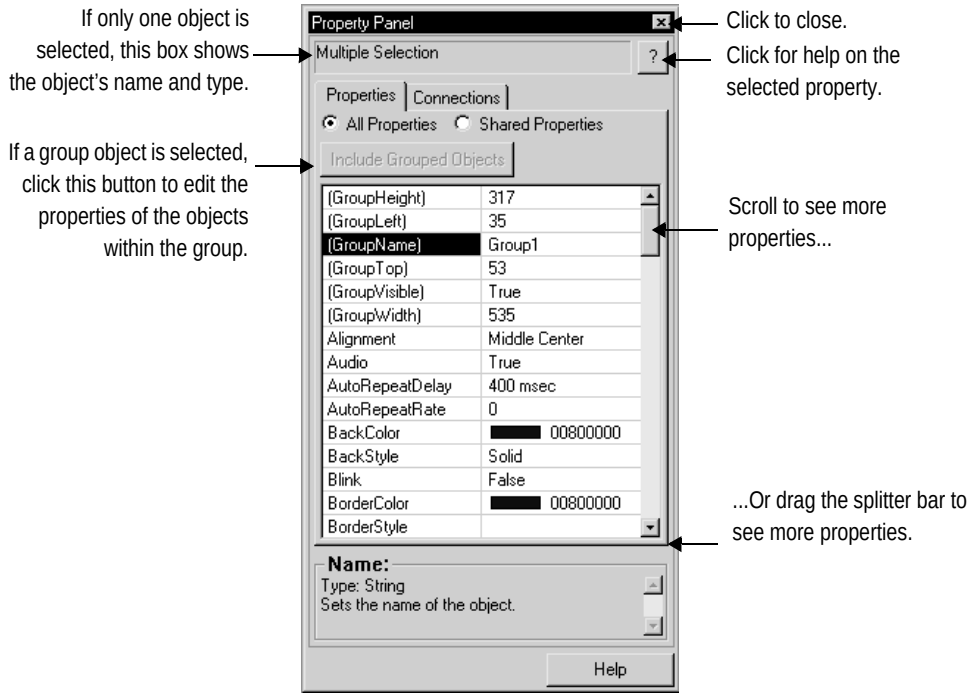
To open the Property Panel, use one of these methods

- On the Graphics toolbar, click the Property Panel tool.
- On the View menu, click Property Panel.
- Right-click an object, and then click Property Panel.
- Right-click an empty area of the display, and then click Property Panel.

You can keep the Property Panel open as you work in the Graphics editor. You can drag the panel's borders to make the Property Panel larger or smaller.

Setting up properties

Use the Property Panel's Properties tab to set up the properties of the selected object or objects.



For more information about using the Property Panel to set up an object's properties, see Help.

Assigning tags and expressions to an object's connections

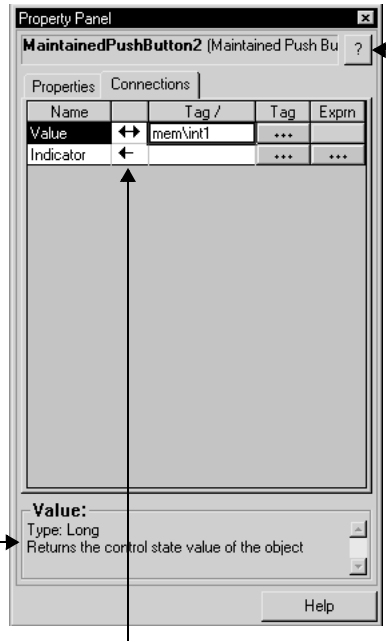
Use the Property Panel's Connections tab to assign tags or expressions to the selected object's connections. If multiple objects are selected the tab is blank, because you can assign tags or expressions to only one object at a time.

How values are updated

The arrows indicate the direction in which the data flows between the connection and the data source:

- A right arrow indicates that the connection sends values to the data source. The connection is a write connection.
- A left arrow indicates that the data source sends values to the connection. The connection is a read connection.

- A double arrow indicates that the data flows in both directions. The connection is a read-write connection.



Click for help on the selected connection.

This box describes the selected connection and indicates the type of data the connection uses.

The arrows show the direction in which data flows between the connection and the data source. If the connection is a read connection (arrow points left), data flows from the data source to the object. If the connection is a write connection (arrow points right), data flows from the object to the data source. If the connection is a read and write connection (double-headed arrow), data flows in both directions.

For more information about using the Property Panel to assign tags and expressions to an object's connections, see Help.

For more information about assigning tags and expressions to objects, see page 20-36.

Coloring objects using the color toolbars

The Foreground Color and Background Color toolbars contain a selection of colors you can assign to objects' color properties.

About color properties

The number of color properties an object has depends on the type of object and how you set it up. For example, a button with states can use up to seven different colors for each state. When you select colors using the color toolbars, some properties are assigned the foreground color and some are assigned the background color. Other color properties, such as Fill color, cannot be assigned using the color toolbars (instead, use the object's Properties dialog box or the Property Panel).

This table lists the color properties you can assign using the color toolbars:

Property	Foreground color	Background color
Back color	No	Yes
Background color	No	Yes
Border color	Yes	No
Caption color	Yes	No
Caption back color	No	Yes
Fore color	Yes	No
Foreground color	Yes	No
Image color	Yes	No
Image back color	No	Yes
Legend color	Yes	No
Needle color	Yes	No
Pattern color	Yes	No

For objects with states, the selected color is applied to the current state's color properties only. In the Property Panel, properties that apply to states have "St_" at the beginning of the property name.

When to select colors using the toolbars

For all the drawing objects except image and panel, you can select colors from the color toolbars before you draw an object (either before or after you click the object's tool).

The other objects are always drawn using their default colors, but you can select the objects and then click the toolbars to change their colors. The toolbars are especially useful for quickly assigning the same colors to multiple objects.

To display a color toolbar

1. On the View menu, select Toolbars, and then click Foreground Colors or Background Colors.

To close a color toolbar

1. On the View menu, select Toolbars, and then click Foreground Colors or Background Colors, or click the toolbar's Close button.

Other methods for assigning colors

You can also assign colors using an object's Properties dialog box or the Property Panel. Use one of these methods if you want to assign separate colors to different foreground or background color properties, or to choose colors that don't appear in the toolbars.

For example, if you want to use a dark blue background color for a button, with a light blue background color for its image label, you must assign the colors separately. Using the toolbar would assign the same color to both properties.

Also use the Properties dialog box or Property Panel to change the default colors for properties that cannot be assigned using the color toolbars.

Naming objects

Objects (and groups of objects) are automatically given a name and number when you create them, for example NumericInputEnable4. If desired, you can assign a more meaningful name to the object, for example Conveyor_speed_input. Each object in a display must have a unique name.

The object name appears in the status bar, Diagnostics List, Property Panel, and Object Explorer in FactoryTalk View Studio, and in diagnostics log messages at run time.

To name an object in the Property Panel

1. Select the object.
2. In the Property Panel, click the Properties tab.
3. Double-click the (Name) row, and then type the name.

   The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

To name an object in its Properties dialog box

1. Double-click the object to open its Properties dialog box.
2. Click the Common tab.
3. In the Name box, type the name.

The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

To name a group object

1. Select the group object.
2. In the Property Panel, click the Properties tab.
3. Double-click the (GroupName) row, and then type the name.

The name must start with a letter, and cannot contain spaces. You can use the underscore character (_).

Testing how objects look in different states

To make sure the different states for an object are set up correctly, you can view them using the States toolbar or the Property Panel.

To open the States toolbar

1. On the View menu, select Toolbars, and then click States.



To view an object's states using the States toolbar

1. Select one or more objects.
2. In the States toolbar, select the state to view.

   If you selected multiple objects, the toolbar shows the states that are common to all the objects.
3. To view the next state, select it in the toolbar or press the Down Arrow key on your keyboard.

   You can also use these keys to select the next state to view:

   - To view the previous state, press the Up Arrow key.
   - To view the first state, press the Home key.

- To see the last state, press the End key.

To view an object's states using the Property Panel

1. Select one or more objects.
2. In the Property Panel, click the State property and then select the state to view.
3. To view the next state quickly, double-click the row, or press the Enter key on your keyboard.

Assigning tags and expressions to graphic objects

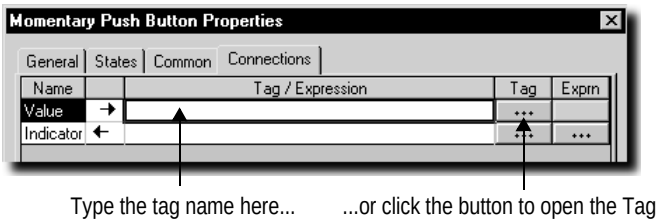
You can assign tags and expressions to many graphic objects, including ActiveX objects (depending, of course, on how the vendor implemented the object). This section describes how to:

- assign tags to graphic objects.
- use expressions to manipulate tag values.
- replace tags using tag substitution.
- use tag placeholders so the same display can be used with different sets of tags.

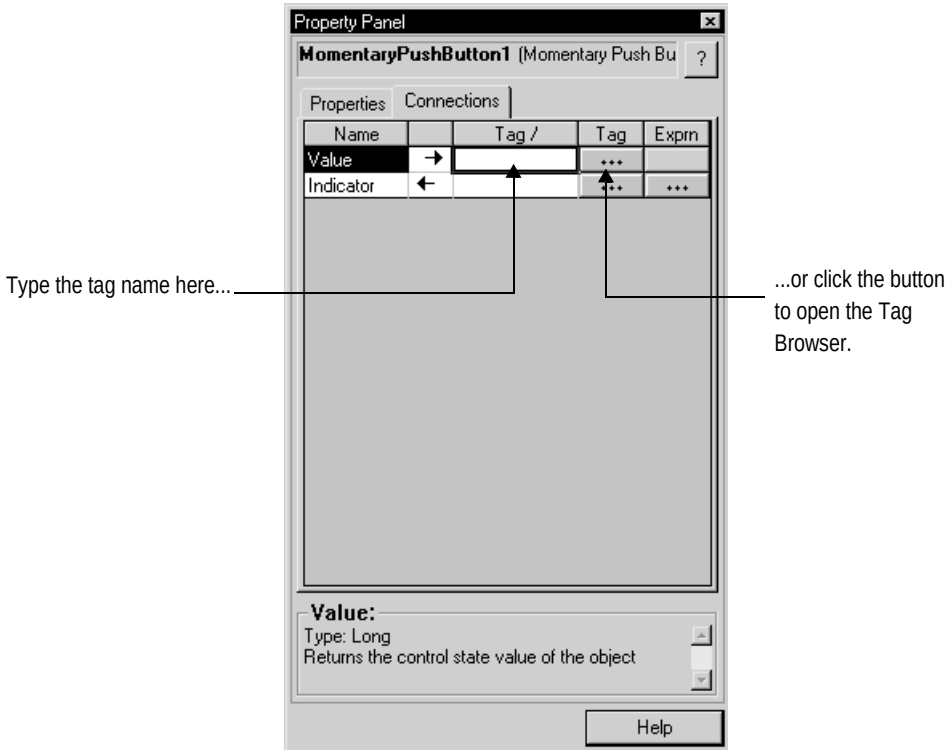
Assigning tags

To assign tags to a graphic object, use one of these methods:

- Double-click the object to open the object's Properties dialog box, and then assign tags in the Connections tab.



- Select the object and then assign tags in the Connections tab of the Property Panel.



- Select the object, and then on the Edit menu click Connections. Assign tags in the Connections tab of the Properties dialog box.
- Right-click the object, and then click Connections. Assign tags in the Connections tab of the Properties dialog box.

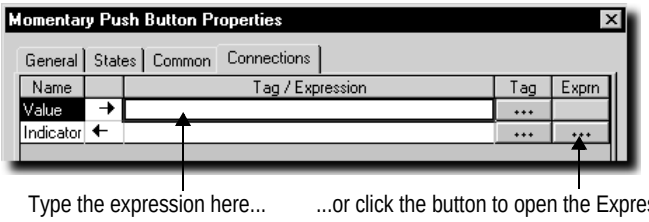
For information about:

- using the Tag Browser, see page 6-6.
- using the Properties dialog box, see page 20-26.
- using the Property Panel, see page 20-29.

Using expressions to manipulate tag values

Many of the connections to which you can assign a tag also permit the use of expressions to perform logical or mathematical calculations on tag values. If you assign an expression, FactoryTalk View monitors the expression value rather than the original tag value. For example, your machine might send values to the data source in the form of temperature in degrees Celsius. You could use an expression to convert the value to degrees Fahrenheit, and then monitor the expression result rather than the original tag value.

If you can assign an expression, a Browse button is present in the Exprn column in the Connections tab.



To specify an expression, do one of the following

- In the "Tag / Expression" column, type the expression.
- In the Exprn column click the Browse button and then create an expression in the Expression editor. Use this method if you want to check the expression syntax, or to use multiple lines for the expression.

For more information about expressions, see Chapter 23.

Replacing tags using tag substitution

You can replace tags assigned to the graphic objects in your display by using tag substitution. You can also replace the tags used in expressions assigned to graphic objects.

For example, if you assign a tag called HoistHeight to multiple objects in the display, and then decide to use the tag Hoist_height instead, you can use tag substitution to quickly replace the old tag with the new tag.

You can replace:

- a tag name (with or without folder names).
- a folder name.
- the text in an expression.
- tags contained in embedded variables.

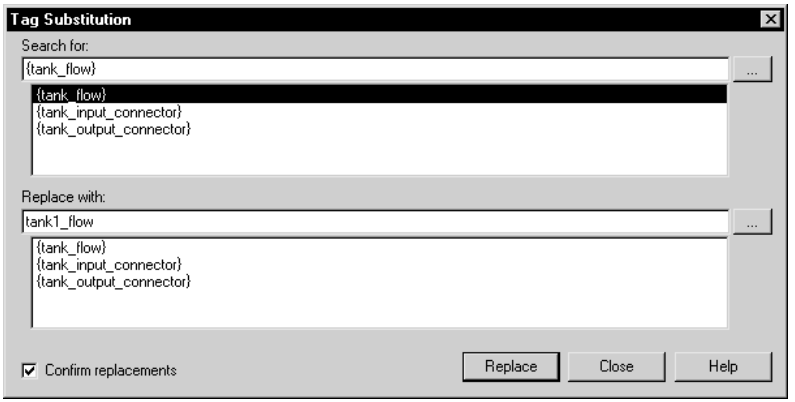
To replace tags

1. Select one or more objects.

   To select all the objects in the display, on the Edit menu, click Select All.

2. On the Edit menu, click Tag Substitution.

   To replace the tags for a single object, you can right-click it and then click Tag Substitution.



For details about the options in the Tag Substitution dialog box, see Help.

Using tag placeholders

Tag placeholders provide a way to use one graphic display to represent a number of similar operations.

For example, suppose you are creating displays for a plant that cans corn and peas. The machinery used in both processes is identical. Instead of creating two displays and specifying corn-related tags in one display and pea-related tags in another, you can create one display and not specify any tag names. Where a tag name is required, type a tag placeholder.

You can use tag placeholders wherever you would normally assign a tag to an object, including in expressions and embedded variables. You can also use tag placeholders in the expressions you create to animate objects.

You can also use tag placeholders with global objects. For more information, see Chapter 25.

You can use tag placeholders in:

- the graphic display that opens when the application is first run.

- graphic displays that are opened using a goto display button.
- graphic displays that are opened using a display list selector.

Use parameter files to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters. For more information about using parameter files and global object parameters, see Chapter 25.

Creating tag placeholders

A tag placeholder is the cross-hatch character (#) followed by a number from 1 to 500.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

You can assign tag placeholders in:

- the Connections tab of an object's Properties dialog box.
- the Connections tab of the Property Panel.
- the Expression box in the Animation dialog box.
- anywhere that you can insert an embedded variable. For information about embedded variables, see Chapter 24.

To create a tag placeholder

1. Type the cross-hatch character followed by a number (no space in between). For example, #1.

Performing basic operations on objects

Once you have drawn an object, you can select the object and work with it. You can:

- move objects.
- copy objects.
- duplicate objects.
- resize objects.
- reshape drawing objects.
- delete objects.

Moving objects

You can move objects using the mouse or the keyboard. The keys give you fine positioning, allowing you to move objects in small increments. You can also use the grid to position objects precisely.

Another option is to position an object using the object's Top and Left properties in the Property Panel. For information about using the Property Panel, see page 20-29.

You can also specify an object's position using the Common tab in the object's Properties dialog box. For more information, see page page 21-1.

Once you've moved objects into position, you might want to align other objects with them, or lock them into place. For information about aligning objects, see page 20-50. For information about locking objects into position, see page 20-56.

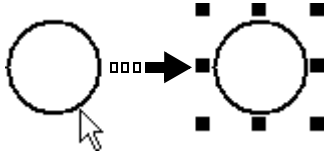
To automatically align objects to the grid as you move them

1. On the View menu, select Snap On. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see page 19-8.

To move objects by dragging with the mouse

1. Select one or more objects.
2. Place the pointer on an object (not on the edge or on the handles).
3. Drag the objects to the desired position.



Select the object. Drag the object to the desired position.

If you selected several objects, dragging one of the objects moves all the selected objects. The objects maintain their position relative to each other.

To move objects in small increments using the keyboard

1. Select one or more objects.
2. Place the pointer on the object, not on its edge or handle.
3. Hold down Shift while you press an arrow key.

To adjust the amount of the increment, first hold down the Shift key and press the + or - keys on the keyboard's numeric keypad.

4. Release the Shift key when the object is in place.

Copying objects

To copy objects, you can:

- drag and drop objects in the same display.
- drag and drop objects between displays, or from a graphic library to a display.
- copy and paste objects.

When an object is copied, any animation attached to the object is also copied. If a group is copied, the new copy of the group can be ungrouped to individual objects, just like the original.

Copying objects with multiple languages

If an object has multiple language strings set up, copying the object copies all the languages. If the object is pasted into an application with different languages, only the strings for languages that are used by the application are pasted. If the new application has languages that are not set up for the object, those language strings are undefined and will be displayed with single question marks.

For more information about setting up multiple languages, see Chapter 12.

To copy objects in the same display

1. Select one or more objects.
2. Drag the object, and then press Ctrl.

   When you press Ctrl, a plus sign is added to the cursor.
3. When the object is where you want it, release the mouse button and then the Ctrl key.

   A new copy of the object is created.

   If you selected several objects, dragging one of the objects copies all the selected objects. The objects maintain their position relative to each other.

To drag objects between displays

1. Open both displays (or a graphic library and a display).
2. Position or resize the displays so both are visible.

   For more information, see page 19-13.

3. Select one or more objects.
4. Click the selected object and drag it to the new display.

   If you selected several objects, dragging one of the objects copies all the selected objects. The objects maintain their position relative to each other.

Copying and pasting objects

You can cut, copy, or paste objects using the menu items on the Edit menu or the buttons on the toolbar.

Once you cut or copy an object, you can paste it anywhere in the drawing area of:

- the same graphic display.
- a graphic display in the same or a different application.
- a graphic library in the same or a different application.

To cut or copy objects



Cut tool



Copy tool

1. Select one or more objects.
2. On the Edit menu, click Cut or Copy, or click the Cut or Copy tool on the Graphics toolbar.
   - To remove the original object, click Cut.
   - To retain the original object, click Copy.

To paste objects



Paste tool

1. Click in the display or library to paste to.
2. On the Edit menu, click Paste, or click the Paste tool on the Graphics toolbar.

Duplicating objects

When an object is duplicated, any animation attached to the object is also duplicated. If a group is duplicated, the new copy of the group can be ungrouped to individual objects.

To duplicate objects

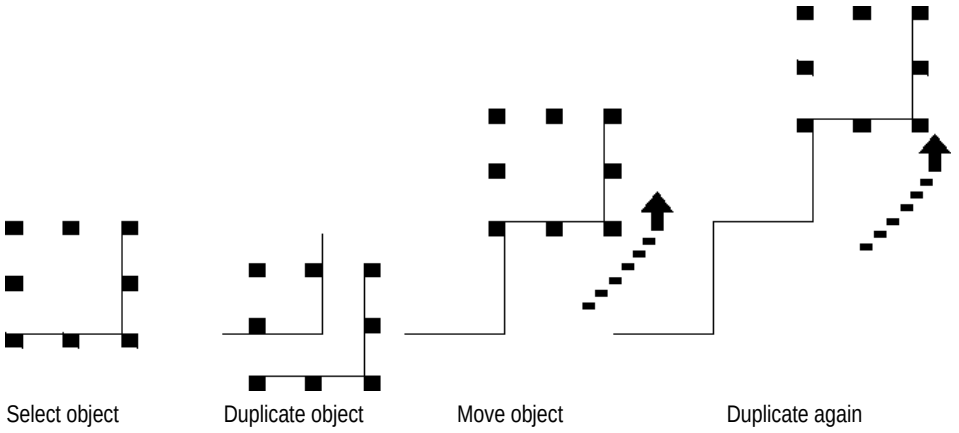


Duplicate tool

1. Select one or more objects.
2. On the Edit menu, click Duplicate, or click the Duplicate tool on the Graphics toolbar.

   The duplicated object is placed slightly offset from the original.

Duplicate also duplicates actions. For example, if you duplicate an object, move it, and then duplicate it again, the second Duplicate will, in one step, duplicate and move the object. This is useful for creating a series of objects with an equal distance between them.



Duplicate works until you deselect the object.

Resizing objects

You can resize objects using the mouse or using the keyboard. The keys let you resize objects in small increments. You can also use the grid to resize objects precisely.

Another option is to size an object using the object's Height and Width properties in the Property Panel. This method is especially useful for quickly resizing multiple objects to the same size. For information about using the Property Panel, see page 20-29.

You can also specify an object's size using the Common tab in the object's Properties dialog box. For more information, see page 21-1.

When you resize text objects, if you have selected the Size to fit option, the font size is adjusted to fit the new object size as closely as possible.

To automatically align objects to the grid as you resize them

1. On the View menu, select Snap On. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see page 19-8.

To resize an object using the mouse

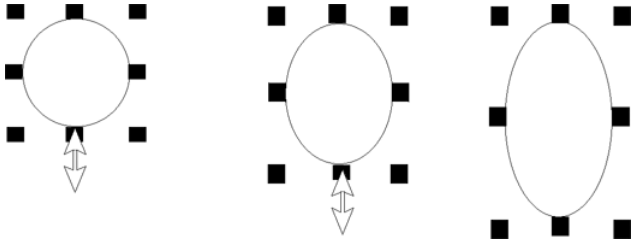
1. Select the object.

2. Place the pointer on one of the handles.

   A double arrow appears.

3. Drag the handle until the object is the desired size or shape.

   Drag a side handle to change width or height, or a corner handle to change both.



   For perfect circles and squares, press Ctrl and hold the key down while you drag a corner handle.

   To maintain the object's original proportions (width to height), press Shift and hold the key down while you drag a corner handle.

To resize an object in small increments using the keyboard

1. Select the object.
2. Place the pointer on one of the handles.

   A double arrow appears.

3. Hold down Shift and press an arrow key until the object is the desired size.

   To adjust the amount of the increment, first hold down the Shift key and press the + or - keys on the keyboard's numeric keypad.

Reshaping drawing objects

You can reshape arcs, lines, polygons, polylines, rectangles, rounded rectangles, and wedges.

To reshape lines, rectangles, polylines, and polygons

1. Select the object you want to reshape.
2. Click the Polygon tool, or right-click the object and then click Edit.



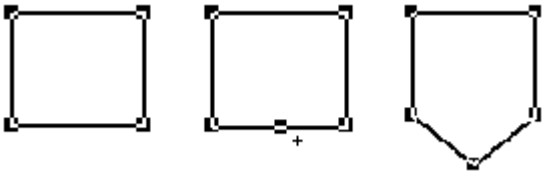
Polygon tool

   The cursor changes to the Polygon tool, and handles appear on the object.

3. Move the cursor over any line or corner of the object.

   A handle with a cross-hair appears.

4. Drag the handle until the object is the desired shape.



Dragging from a point along the line (between corners) creates a new angle between the two corners.

5. To delete an angle, position the pointer at the tip of the angle, and then press Delete.

To reshape arcs and wedges

1. Select the object you want to reshape.
2. Click the Arc or Wedge tool, or right-click the object and then click Edit.



Arc tool

The cursor changes to the Arc or Wedge tool, and handles appear on the object.

3. Place the pointer on one of the handles.



Wedge tool

A cross-hair appears.

4. Drag the handle until the object is the desired shape.

To reshape rounded rectangles

1. Select the rounded rectangle.
2. Click the Rounded Rectangle tool, or right-click the object and then click Edit.



Rounded rectangle tool

A handle appears inside the rounded rectangle.

3. Click the handle and drag inward to increase roundedness, or outward to decrease roundedness.

Deleting objects



Undo tool

If you accidentally delete an object, use the Undo tool to restore it.

To delete objects

1. Select one or more objects.
2. On the Edit menu, click Delete, or press Delete on the keyboard.

To delete all the objects in the display

1. On the Edit menu, click Clear All.

Working with groups of objects

Grouping and ungrouping objects

Grouping combines several objects into a single object so you can manipulate them as a single object. Grouping is useful for keeping objects in the same position relative to each other. You can cut, copy, and paste groups, arrange the group as a single object relative to other objects, and apply the same properties to all the members of the group at once.

You can attach animation to a group, and any animation attached to individual objects in the group remains active. The group animation generally takes precedence over the animation of individual objects within the group. For more information, see page 22-14.

Deleting a group deletes all individual objects in the group.

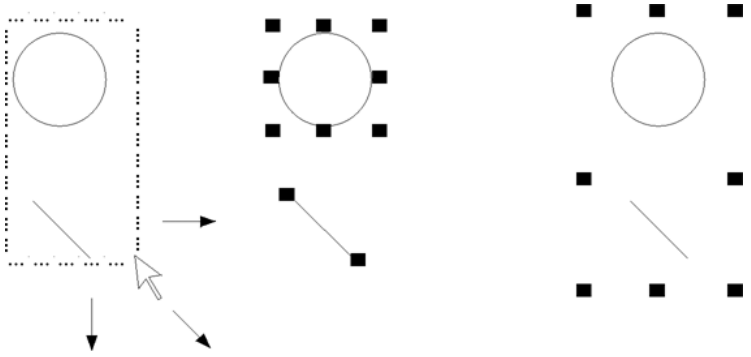
To group objects

1. Select all the objects you want grouped.
2. On the Arrange menu, click Group, or on the Graphics toolbar click the Group tool.



Group tool

The handles around each object are replaced by a set of handles around the group.



Drag the mouse to select the objects.

Group them.

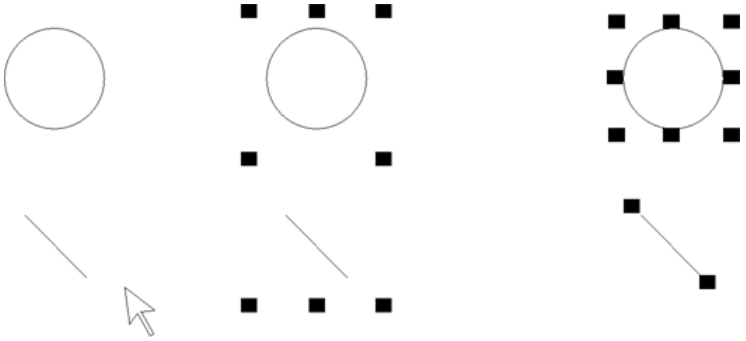
To ungroup objects

1. Select the group of objects to ungroup.
2. On the Arrange menu, click Ungroup, or on the Graphics toolbar click the Ungroup tool.



Ungroup tool

The handles around the group are replaced with handles around each object.



Ungrouping deletes any animation attached to the group, because the group no longer exists. However, animation attached to the individual objects that were in the group remains active.

Editing groups of objects

Edit a group the same way you would edit an individual object. You can:

- use the Property Panel to apply the same properties to all the members of the group at once. For example, change the line width of all objects in the group to 2 pixels.

  For information about using the Property Panel, see page 20-29.

- use the toolbars to apply the same pattern style, background style, foreground colors, and background colors to all the members of the group.
- for objects with states, use the States toolbar to cycle through the states and apply the same properties to the states for each object in the group at once. When you select a group containing objects with states, only the states that are common to all objects in the group appear in the toolbar.

  For more information about using the States toolbar, see page 20-35.

Editing objects within a group

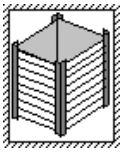
You can also edit individual objects within the group without breaking the group, which is particularly useful when you have animation attached to the group.

To edit objects within a group

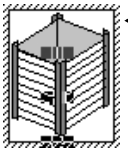
1. Double-click the grouped object, or right-click it and then click Edit. A hatched border appears around the group.

When the hatched border is around the group, you are in group edit mode. In this mode, you can select individual objects in the group and modify them.

You can also add new objects to the group.



Double-click to edit the group. A hatched border appears. This border indicates that this is a group.



Click again. Handles appear. The handles show the individual object selected within the group.

2. To select an individual object (or a group) in the group, click it. You can also use the Object Explorer to select objects within the group.

   The status bar and Object Explorer indicate which object or group is selected.

3. Make your changes to the object.

   You can change the selected object's shape, size, or position, or use the object's Properties dialog box or the Property Panel to edit the object's properties.

4. If desired, create new objects inside the hatched border.

5. To stop editing, click outside the group.

Arranging objects

You can arrange objects (or groups of objects) in a number of ways. You can:

- layer objects by moving them in front of or behind other objects.
- align objects with each other and with the grid.
- space objects horizontally or vertically.
- flip drawing objects horizontally or vertically.
- rotate drawing objects.
- lock objects into position.

Layering objects

You can layer objects (or groups of objects) on top of each other. Objects are layered in the order they are created, with the most recently created object on top. Change the layer order with Send to Back and Bring to Front.

Send to Back moves the selected object to the bottom layer.

Bring to Front moves the selected object to the top layer.

To bring an object to the front

1. Select the object.

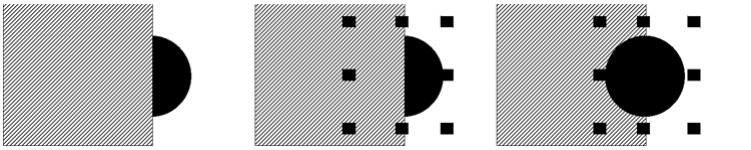
   To select an object that's behind another object, place your pointer on the front object, click once, pause, and then click again. Do not double-click and do not move the mouse.

   You can also select a concealed object easily by clicking the object in the Object Explorer.

2. On the Arrange menu, click Bring to Front, or click the Bring to Front tool.



Bring to Front tool



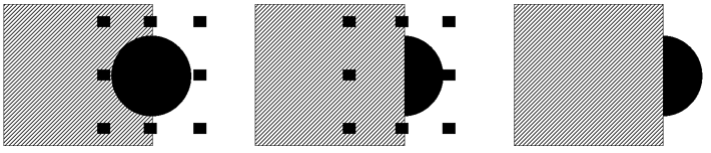
Select the object from behind. Bring the object to front.

To send an object to the back

1. Select an object.
2. On the Arrange menu, click Send to Back, or click the Send to Back tool.



Send to Back tool



Select the object. Send the object to back.

Aligning objects

You can align objects (or groups of objects) with each other and with the grid.

To align objects

1. Select the objects you want to align.

2. On the Arrange menu, click the appropriate menu item, or click a tool on the Alignment toolbar:

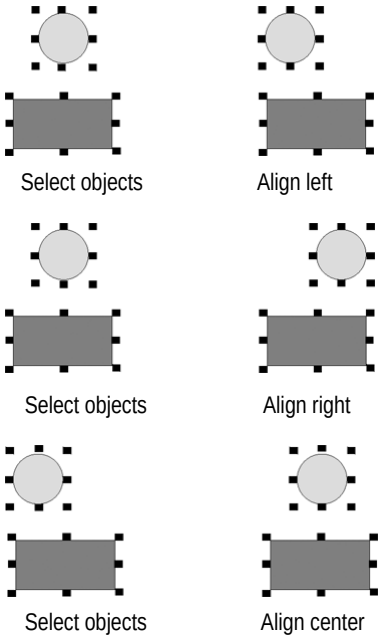
This button or menu item	Aligns selected objects with the
 Align Left	Left-most selected object
 Align Center	Horizontal center of all selected objects
 Align Right	Right-most selected object
 Align Top	Top-most selected object
 Align Middle	Vertical center of all selected objects
 Align Bottom	Bottom-most selected object
 Align Center Points	Center of all selected objects
 Align to Grid	Grid

To automatically align objects to the grid as you create or move them

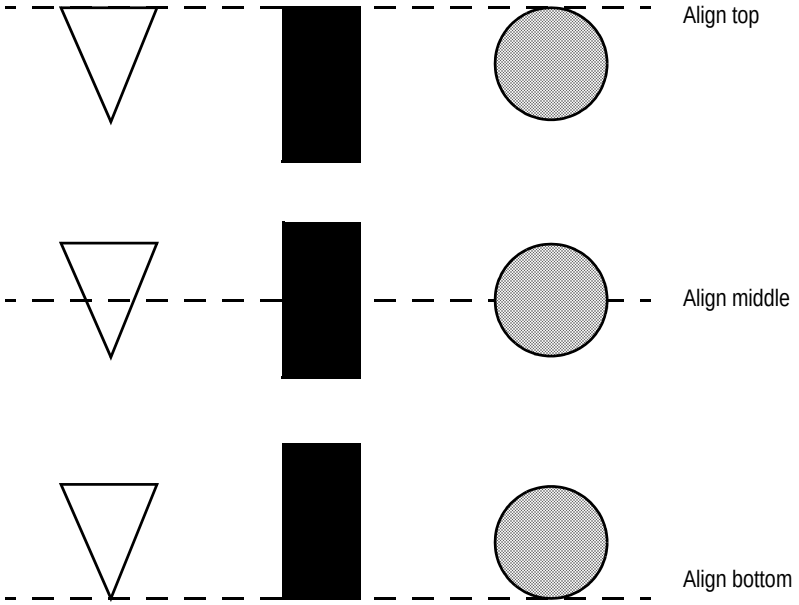
1. On the View menu, select Snap On. A check mark appears beside the menu item when the option is selected.

For information about setting up the grid, see page 19-8.

Examples: Aligning objects left, right, and center



Examples: Aligning objects top, middle, and bottom



Spacing objects

With Space Vertical and Space Horizontal, objects (or groups of objects) are moved vertically or horizontally to have the same amount of space from the center point of each object.

To space objects

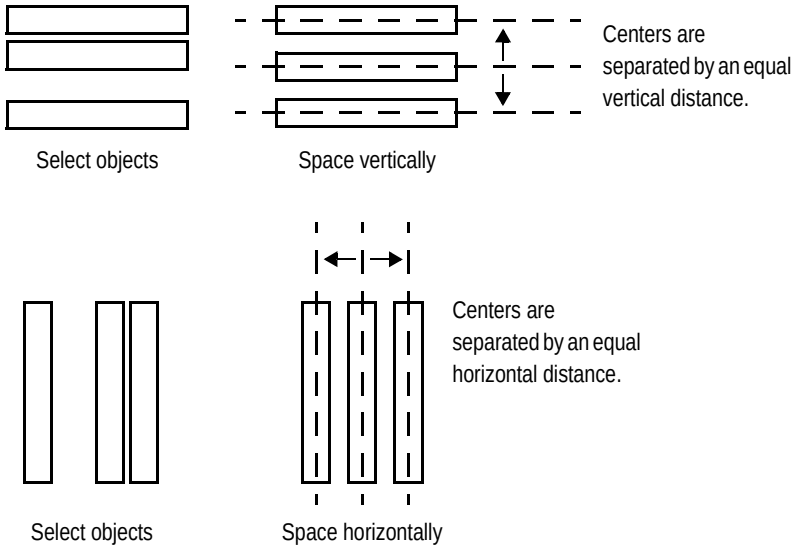
1. Select the objects you want to space.
2. On the Arrange menu, click a Space menu item, or click a tool on the Graphics toolbar:


Space Horizontal tool


Space Vertical tool

This tool or menu item	Does this
Space Horizontal	Places the centers of the selected objects an equal distance apart horizontally.
Space Vertical	Places the centers of the selected objects an equal distance apart vertically.

Examples: Spacing objects vertically and horizontally



Flipping drawing objects

Flipping an object creates a mirror image of the object. You can flip all the drawing objects (or groups of drawing objects) except text, images, and panels.

To flip a drawing object

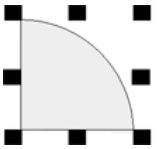
1. Click the object.
2. On the Arrange menu, click a Flip menu item, or click a tool on the Graphics toolbar:

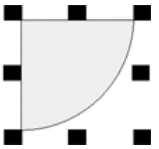

Flip Vertical

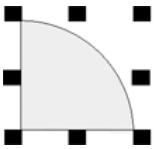

Flip Horizontal

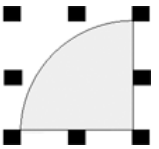
This tool or menu item	Flips selected objects
Flip Vertical	Top to bottom (upside-down)
Flip Horizontal	Left to right

Examples: Flipping drawing objects vertically and horizontally


Select object


Flip vertical


Select object


Flip horizontal

Rotating drawing objects

You can rotate all the drawing objects (or groups of drawing objects) except images, panels, and rounded rectangles.

You can attach rotation animation to the same drawing objects. With rotation animation, the object rotates around an anchor point to indicate a tag's value at run time. For details about rotation animation, see page 22-13.

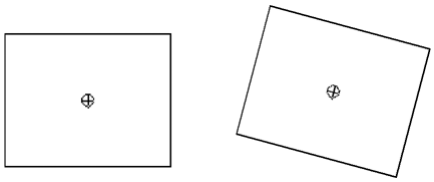
When you rotate text, it rotates around the anchor point but the text itself remains upright.

To rotate a drawing object

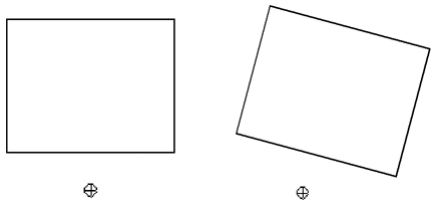

Rotate tool

1. On the Objects menu, click Rotate, or on the Objects toolbar click the Rotate tool.
2. Click the object you want to rotate. A small crosshair circle appears in the middle of the object. This is the anchor point that is used as the center of rotation.

You can place the crosshair inside an object.



You can place the crosshair outside an object.



3. To move the center of rotation, click the cross-hair and drag it to a new anchor position. The anchor can be inside or outside the object.
4. Click an edge of the object and drag in the direction you want to rotate it.

   To rotate the object in five-degree increments, press Ctrl while you drag.
5. When the object is in the desired position, release the mouse button.

Locking objects into position

You can lock graphic objects (or groups of objects) into position by converting them to wallpaper. Once you convert objects to wallpaper, you cannot select or edit them unless you unlock the wallpaper. Wallpaper objects cannot be animated at run time.

If the grid is on, wallpaper objects are positioned behind the grid.

If you just want to lock the objects into place while you're working in the display, unlock the wallpaper when you're finished. If you want to use the wallpaper objects as a background for your display, leave the wallpaper locked.

For more information about creating a background for your display, see page 19-14.

21 Setting up graphic objects

This chapter describes how to set up graphic objects. It describes how to:

- set up objects' spatial properties, names, and visibility.
- work with objects that have states.
- position objects for touch screens.
- assign function keys to buttons.
- use the keyboard to navigate to and select objects at run time.
- link buttons to lists and trends.
- repeat a button's action by holding down the button.
- ensure values are read by the data source before sending new values.
- set up objects.

For objects that are not described in other chapters of the manual, this chapter provides information about how to use the objects.

For information about creating graphic objects, see Chapter 20.

For information about creating and setting up the trend graphic object, see Chapter 28.

For information about creating and setting up the RecipePlus graphic objects, see Chapter 29.

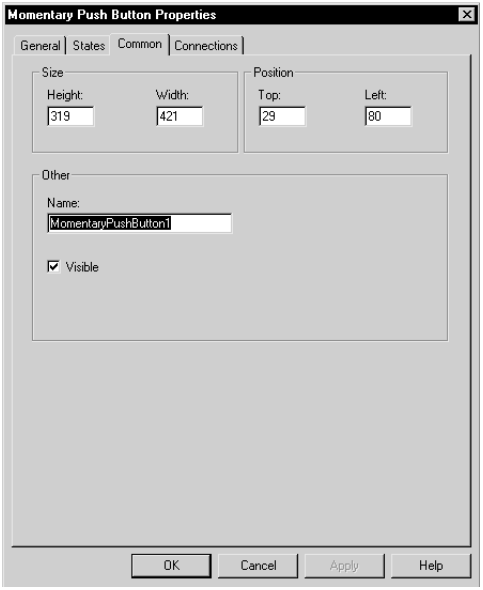
Setting up objects' spatial properties, names, and visibility

Every graphic object has a Common tab in its Properties dialog box. Use the Common tab to set up the following properties for the object:

- height and width
- top and left position
- name
- visibility

For ActiveX® and trend objects, you can also set up these properties:

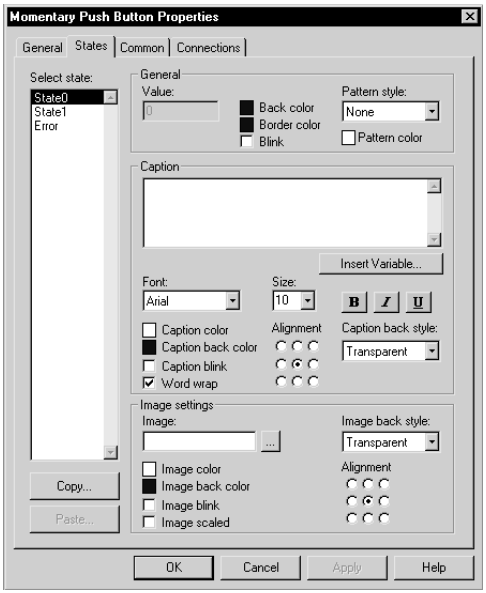
- focus highlight
- key navigation



For details about the options in the Common tab, see Help.

Tips for setting up objects with states

The graphic objects that have states have a States tab in their Properties dialog box.



For details about the options in the States tab, see Help.

Copying and pasting properties from one state to another

Often, you'll want most of the properties to be the same from state to state, with only one or two settings changing to distinguish the different states.

To simplify setting up states when many of the properties are the same, you can copy and paste settings from one state to another. For details, see Help.

Adding and removing states

Some graphic objects have a configurable number of states. For these objects, you can use the Insert State and Delete State buttons in the States tab, to add and remove states without returning to the General tab. The "Number of States" setting on the General tab is automatically updated.

For more information about the Insert State and Delete State buttons, see Help.

Setting up how objects are used at run time

Just as you must provide operators with a way to navigate between displays at run time, you must also make sure that operators can use the objects within the displays. The next sections in this chapter describe how to set up your objects so the operator can:

- press objects using a touch screen.

- use function keys to press buttons when a mouse or touch screen is not available on the runtime computer.
- use the keyboard or keypad to navigate to and select lists, trends, and ActiveX input objects.
- use buttons to work with lists and trends.
- repeat a button's action by pressing and holding it.
- ensure that tag values are read by the data source before sending new values.

Positioning objects for touch screens

If the operator will be using a touch screen at run time, keep these tips in mind when positioning graphic objects in your displays:

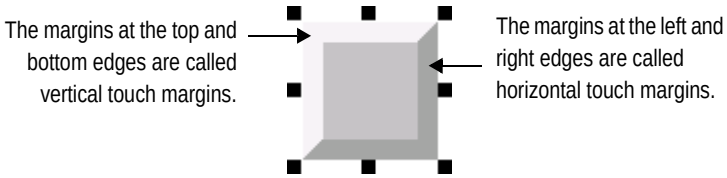
- Don't place important buttons where they'll be blocked by an On Top display. The user can't press a covered button.
- Ensure buttons are large enough for users to touch easily.
- Use touch margins for buttons that are positioned close together, to ensure that the adjacent button is not pressed by mistake.

Using touch margins

Touch margins are touch-insensitive borders inside the button's margin. If the operator presses on the touch margin, the button press is not registered. Touch margins are useful when buttons are positioned close to each other and you want to make sure the operator doesn't press the wrong button by mistake.

You can create touch margins at the top and bottom of the button, at the sides, or on all four sides.

In the illustration below, the button's border and touch margins are the same size, 12 pixels. A button press would be registered only when the darker square in the middle of the button is pressed.


The margins at the top and bottom edges are called vertical touch margins.
The margins at the left and right edges are called horizontal touch margins.

The bounding box

If the object's shape is a circle or ellipse, the touch margin applies to the object's bounding box, not the object's border. The bounding box is an invisible square or rectangle that surrounds the object. When you select the object, the selection handles show the location of the bounding box.



To create touch margins, use one of these methods

- In the General tab of the button's Properties dialog box, type the number of pixels for the touch margins in the Horizontal margin and Vertical margin boxes.
- In the Properties tab of the Property Panel, type the number of pixels for the HorizontalMargin and VerticalMargin properties.

Assigning function keys to buttons

You can assign function keys to the buttons in your displays to allow the operator to press the buttons using the function keys on the runtime terminal (or the function keys on a keyboard, if one is available). You can also assign a function key to the numeric input cursor point.

You can assign up to 34 function keys to each graphic display.

Here are some tips for assigning function keys:

- Where possible, use the same function keys for the same operations in all your graphic displays. For example, if every display contains a goto display button that returns the operator to a graphic display called Main Menu, assign the same function key to this button in each display.
- Include the name of the function key assigned to a button in a caption on the button.

  For buttons with multiple states, include the function key name in the caption for each state, or create a text object to use as a label for the button (so that you don't have to set up the caption for each state), and then group the text and button together.
- If your application will run on a PanelView™ Plus or PanelView™ Plus CE terminal, assign keys that are supported by the runtime terminal. Different sizes of terminals have different function keys.

Function key equivalents

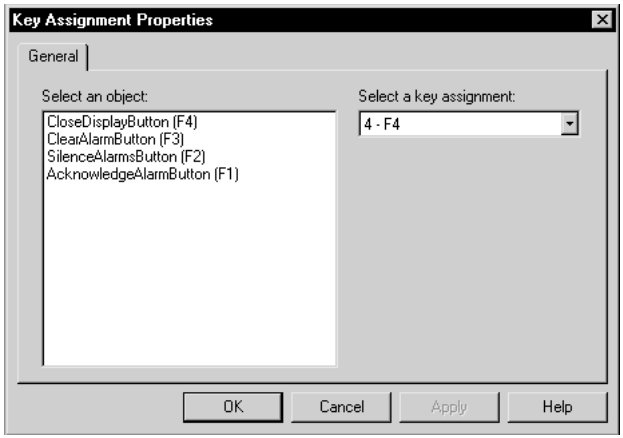
If your application will run on a personal computer, the keyboard contains the function keys F1 to F12 only. The remaining function keys are associated with these key combinations:

For this function key	Use this key combination
F13	Left Shift-F1
F14	Left Shift-F2
F15	Left Shift-F3
F16	Left Shift-F4
F17	Left Shift-F5
F18	Left Shift-F6
F19	Left Shift-F7
F20	Left Shift-F8
K1 - K12	Right Alt-F1 to Right Alt-F12
K13	Right Shift-F1
K14	Right Shift-F2
K15	Right Shift-F3
K16	Right Shift-F4
K17	Right Shift-F5
K18	Right Shift-F6
K19	Right Shift-F7
K20	Right Shift-F8

At run time, the operator presses the key combination to activate the object to which the function key is assigned.

To assign function keys to buttons

1. On the Edit menu, click Key Assignments, or right-click a button and then click Key Assignments.



For details about using the Key Assignment Properties dialog box to assign function keys and change function key assignments, see Help.

Using the keyboard to navigate to and select objects

If a mouse or touch screen is not available on the runtime computer, the operator can use the keys on a keyboard or keypad to select (give focus to) these objects:

- lists: control list selector, piloted control list selector, display list selector, diagnostics list, alarm list, and alarm status list
- alarm banners
- trends
- numeric input enable buttons and string input enable buttons
- numeric input cursor points
- RecipePlus table and selector
- third-party ActiveX input objects

What input focus looks like

The object with focus is surrounded by a highlight box, unless the Disable Highlight When Object has Focus box is selected (in the Display Settings dialog box). You can specify the color of the highlight in this dialog box as well. For more information, see Help.



Focus highlight for ActiveX and trend objects

For ActiveX and trend objects, use the Common tab in the object's Properties dialog box to specify whether or not to display a highlight box. For more information about setting up options on the Common tab, see page 21-1.

If the Disable Highlight When Object has Focus box is checked (in the Display Settings dialog box), that setting overrides the setting you specify in the Common tab.

Using the keys on the keyboard or keypad

When a graphic display opens, the keyboard-navigable object that is closest to the top left corner of the display is selected. The operator can use these keys to move to and select a different object:

Use this key	To do this
Tab	Move from the upper left to the lower right.
Shift+Tab	Move from the lower right to the upper left.
Ctrl+arrow key	Move left, right, up, or down.

Removing objects from and adding objects to the tab sequence

By default, you can use the keys to navigate to all lists, alarm banners, numeric input cursor points, trends, and ActiveX input objects in a display. However, you can turn off key navigation for these objects if desired.

When an object's key navigation is turned off, the operator can still select the object using a mouse or touch screen, if available.

By default, key navigation is turned off for the numeric input enable buttons and string input enable buttons. You can turn on keyboard navigation if you want the operator to use the keyboard to navigate to these objects.

To turn key navigation on or off, use one of these methods

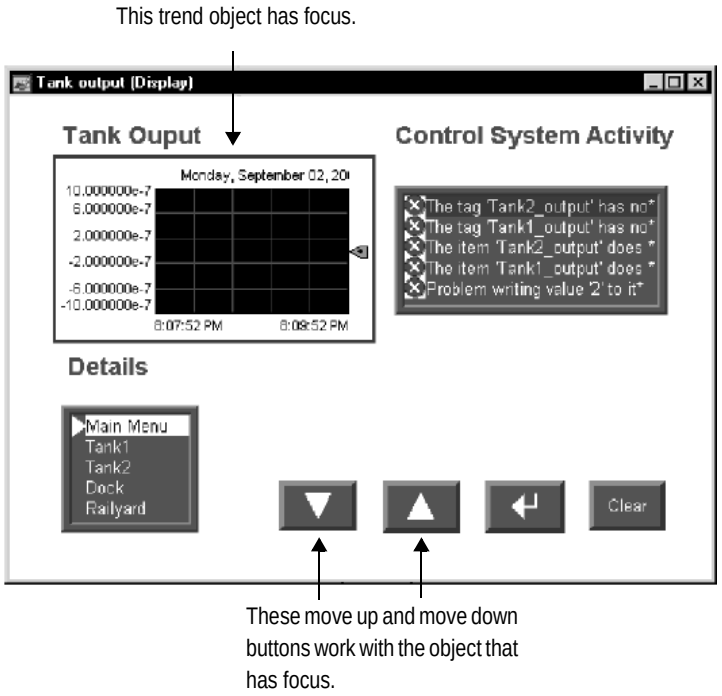
- In the Properties tab of the Property Panel, set the KeyNavigation property to False for off or True for on.
- For list objects, alarm banners, numeric input cursor points, numeric input enable buttons, and string input enable buttons, in the General tab of the object's Properties dialog box, clear the Key navigation box for off. Check the box for on.
- For ActiveX and trend objects, in the Common tab of the object's Properties dialog box, clear the Key navigation box for off. Check the box for on.

Linking buttons to objects

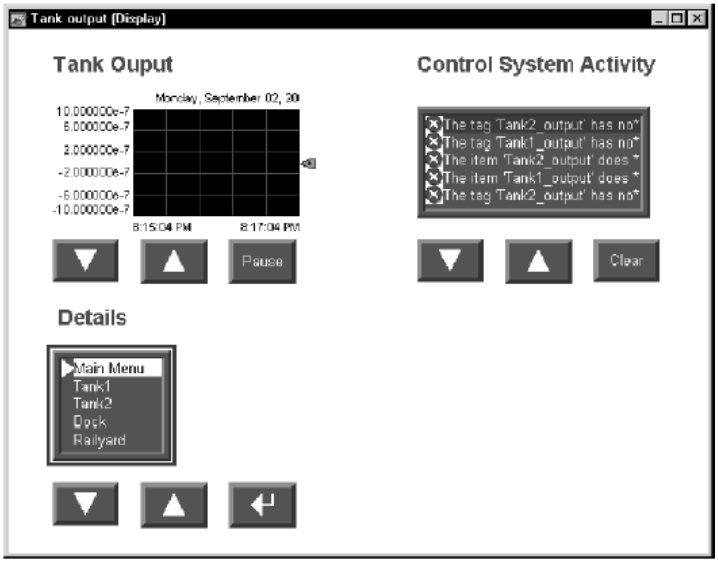
Some FactoryTalk® View buttons can be linked to specific trend, list, recipe, and alarm banner objects, or you can set them up to work with whichever object has focus in the display.

If you are creating small graphic displays that don't have much room, you might prefer to use one set of buttons to work with all the graphic objects (that accept input focus) in the display.

By default, the buttons are set up to work with whichever object has input focus (is selected) in the display.



However, if space isn't a concern, you can create multiple copies of the buttons you want to use and link them to specific objects. One benefit of linking a button to a specific object is that the operator doesn't have to select the object before pressing the button. Another benefit is that you can position the buttons close to the specified object, making it easier for the operator to understand which buttons work with which objects.



Each object has its own set of buttons.

To link a button to a specific object using the button's Properties dialog box

1. Double-click the button to open its Properties dialog box.
2. In the General tab, in the "Send press to" box, select Linked Object.
3. To select from a list of all the objects in the display that you can link the button to, click the Browse button next to the Linked object box.
4. Click the name of the object to link the button to, and then click OK.
5. Click OK to close the button's Properties dialog box.

To link a button to a specific object using the Property Panel

1. In the Properties tab, specify the SendPressTo and LinkedObject properties.

Once you have linked buttons to an object, you might want to turn off the object's key navigation, since this option is no longer needed. For details, see page 21-9.

Repeating a button's action by holding down the button

To repeat a button's action by pressing and holding it, set up auto repeat for the button. If you set up auto repeat, when the operator presses and holds down the button, repeated button presses are registered until the operator releases the button.

A button press occurs when the operator clicks an object with the mouse, presses it on a touch screen, or presses the function key associated with the object. Auto repeat works with all these methods of pressing buttons.

You can use auto repeat with these buttons:

For this button	Each button press does this
Multistate push	Sends the value for the next state to the data source. External changes to the Value connection are not recognized when the button is in auto repeat mode.
Ramp	Sends the new ramped value to the data source. External changes to the Value connection are not recognized when the button is in auto repeat mode.
Move up	Moves the highlight up an item in the list, recipe selector, or recipe table, scrolls up in the trend, or ramps the numeric input cursor point or numeric input enable button value.
Move down	Moves the highlight down an item in the list, recipe selector, or recipe table, scrolls down in the trend, or ramps the numeric input cursor point or numeric input enable button value.
Page up	Moves the highlight up a page in the list, recipe selector, or recipe table.
Page down	Moves the highlight down a page in the list, recipe selector, or recipe table.
Move left	Scrolls the trend to the left.
Move right	Scrolls the trend to the right.

For each button that uses auto repeat, you can specify these properties:

- Auto repeat rate—the number of times per second a button press is registered when the button goes into auto repeat mode. The default rate is 0, which means that auto repeat is turned off.
- Auto repeat delay—the length of time the button has to be pressed and held down before auto repeat starts.

To set up auto repeat for a button, use one of these methods

- In the Timing tab of the button's Properties dialog box, specify the Auto repeat rate and Auto repeat delay properties.
- In the Properties tab of the Property Panel, specify the AutoRepeatRate and AutoRepeatDelay properties.

Ensuring values are read by the data source before sending new values

To ensure a value is read by the data source before the operator sends a new value, use Enter key handshaking. While Enter key handshaking is in effect for an object, the operator cannot send a new value to the object's Value connection.

You can use Enter key handshaking with these graphic objects:

- control list selector
- piloted control list selector
- numeric input enable button
- string input enable button
- numeric input cursor point

If Enter key handshaking is in effect for one of these objects, the operator can still provide input to other objects in the graphic display.

If the graphic display is closed while Enter key handshaking is in effect, the handshaking is cancelled.

Methods of Enter key handshaking

Enter key handshaking works by setting the object's Enter connection to 1. As long as the Enter connection is set to 1, new values cannot be sent to the Value connection. How the Enter connection is reset to 0 depends on how you set up Enter key handshaking.

There are two ways you can use Enter key handshaking:

- to hold the value at the data source for a specific period of time.
- to hold the value at the data source until the data source notifies FactoryTalk View that the value has been read.

Choose the method that best suits your application needs and communication system.

Holding the value for a specific period of time

To set up an object's Enter key handshaking so that the value at the Value connection is held for a specific period of time, assign a tag to the Enter connection and specify the Enter key hold time. You can also specify an Enter key control delay, if desired.

How handshaking works

This method of Enter key handshaking works as follows:

1. When the operator presses the Enter button, the value is sent to the Value connection and the "Enter key control delay" timer begins timing. (The use of a delay is optional.)
2. If you specify an Enter key control delay, when the time is up, the Enter connection is set to 1. If you don't use the delay, the Enter connection is set to 1 as soon as the operator presses Enter.

   As long as the Enter connection is set to 1, the operator cannot send new values to the data source.
3. When the Enter connection is set to 1, the "Enter key hold time" timer begins timing.
4. When the Enter key hold time has expired, the Enter connection is reset to 0 and the operator can send a new value to the Value connection.

To set up Enter key handshaking to hold the value for a specific period of time

1. In Timing tab of the object's Properties dialog box, specify the Enter key control delay (optional) and Enter key hold time properties.
2. In the Connections tab, assign a tag to the Enter connection. A digital tag is recommended (either an HMI tag or a data server tag).

You can also use the Property Panel to specify the properties and assign a tag to the Enter connection.

Holding the value until the data source acknowledges that it has read the value

To set up an object's Enter key handshaking so that the value at the Value connection is held until the data source notifies FactoryTalk View that it has read the value, use two connections: the Enter connection and the Enter handshake connection.

Instead of using an Enter key hold time, specify an Enter key handshake time. You must also specify the Handshake reset type. You can use an Enter key control delay, if desired.

How the Handshake reset type works

How the Enter handshake connection resets the Enter connection depends on which Handshake reset type you select:

With this handshake reset type	The Enter connection is set to 0 when
Non-zero Value	The Enter handshake connection has a non-zero value. If the Enter handshake connection already has a non-zero value when the value is sent to the Value connection (or when the Enter key control delay has expired, if the delay is used), then the Enter connection is not set to 1, and Enter key handshaking does not take place.
Zero to Non-zero transition	The Enter handshake connection changes from 0 to a non-zero value.

Set up the data source to send a non-zero value to the Enter handshake connection when it has read the new value at the Value connection, and then to reset the Enter handshake connection to 0.

If the Enter key handshake time expires before the Enter handshake connection resets the Enter connection, an error message is sent to FactoryTalk® Diagnostics.

How handshaking works

If you use the Enter handshake connection, handshaking works like this:

1. When the operator presses the Enter button, the value is sent to the Value connection and the “Enter key control delay” timer begins timing. (The use of a delay is optional.)
2. If you specify an Enter key control delay, when the time is up, the Enter connection is set to 1. If you don’t use the delay, the Enter connection is set to 1 as soon as the operator presses Enter.

   As long as the Enter connection is set to 1, the operator cannot send new values to the data source.

   If the Handshake reset type is Non-zero Value, the Enter handshake connection must be 0 when the delay expires in order to set the Enter connection to 1.
3. When the Enter connection is set to 1, the “Enter key handshake time” timer begins timing.
4. The Enter connection remains set until the Enter key handshake time expires or until reset by the Enter handshake connection, whichever happens first.

5. The Enter connection is reset to 0 and the operator can send a new value to the Value connection.

To set up Enter key handshaking to hold the value until the data source has read it

1. In the Timing tab of the object's Properties dialog box, specify these properties:
   - Enter key control delay (optional)
   - Enter key handshake time
   - Handshake reset type
2. In the Connections tab, assign these connections:
   - Enter—assign a tag. A digital tag is recommended (either an HMI tag or a data server tag).
   - Enter handshake—assign a tag or expression.
3. Set up the data source to send a non-zero value to the Enter handshake connection when it has read the new value at the Value connection, and then to reset the Enter handshake connection to 0.

You can also use the Property Panel to specify the properties and assign tags to the connections.

Time, date, and number formats for graphic objects

Graphic objects use the time, date, and number format of the current application language. For example, if the application language uses a comma for the decimal symbol, floating-point values displayed in graphic objects uses a comma for the decimal symbol.

For information about using multiple languages, see Chapter 12.

Setting up buttons

Many of the button graphic objects are set up the same way. The instructions in this section apply to these buttons:

Acknowledge alarm	Logout
Acknowledge all alarms	Move down
Alarm status mode	Move left
Backspace	Move right
Clear alarm banner	Move up

Clear alarm history	Next pen
Diagnostics clear	Page down
Diagnostics clear all	Page up
Display print	Password
End	Pause
Enter	RecipePlus
Goto configure mode	Reset alarm status
Home	Return to display
Information acknowledge	Shutdown
Language switch	Silence alarms
Login	Sort alarms

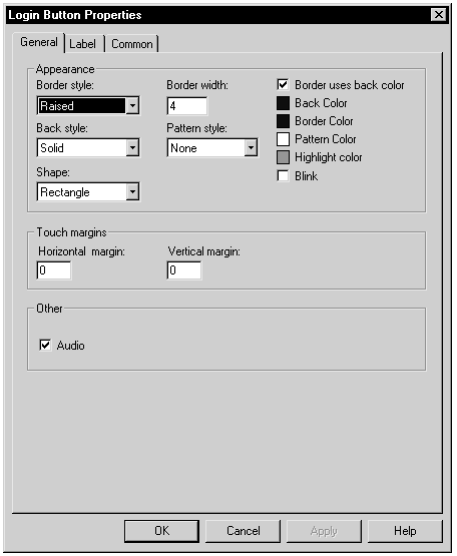
You can set up any combination of these buttons in a graphic display. For example, you might want to put a login, logout, and shutdown button in the same graphic display.

Create and then set up each button separately.

The illustration shows the login button. Some buttons have different properties on the General tab, but you can use these instructions to set up any of the buttons in the list. For example, on the General tab for the acknowledge all alarms button and the clear alarm history button, you can specify a subset of alarms to which the button's action applies. For details about filtering alarm triggers, see page 9-5.

To set up a button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the button looks like at run time. For some buttons, you can specify whether to link the button to a specific object.
   - In the Label tab, specify what text or image to display on the button.
   - In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 21-13.

     The Timing tab is available only for the move up, move down, move left, move right, page up, and page down buttons.
   - In the Common tab, specify the button's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Buttons described later in the chapter

Information about setting up these buttons is described later in the chapter:

Close display	Multistate push
Goto display	Numeric input enable
Interlocked push	Print alarm history
Latched push	Print alarm status
Macro	Ramp
Maintained push	String input enable
Momentary push	

How to use push buttons

Push buttons start or stop processes or actions by changing tag values.

Never use push buttons for emergency stops. Emergency stop buttons must always be hard-wired.

You can use different kinds of push buttons, depending on what kind of machinery you are running or process you are controlling:

Momentary push buttons change a tag to one value when the button is pressed, and another value when the button is released. The machine is on only while the button is held down. When the button is released, the machine turns off. Momentary push buttons are useful for jogging a motor, and they can be set up to start and stop a machine or process.

Maintained push buttons toggle between two values. This type of button is useful for changing a setting within a machine or process, but not for starting the machine or process. For example, use the maintained push button for changing modes, such as Auto to Manual, or Metric to Imperial.

Latched push buttons latch in the on position, and must be unlatched by another button or process to return to the off position. This type of button is useful for starting a machine or process.

Multistate push buttons allow an operator to cycle through multiple options consecutively, using a single button. The current state of a process or operation is displayed on the button by a different color, caption, or image for each state.

Interlocked push buttons work in groups, and share the same tag. Pressing one button cancels another. Although interlocked push buttons work as a group, you add them to the display one at a time.

Ramp buttons increase or decrease the value of a tag by either an integer or floating-point value. You can use two ramp buttons together to create an increment/decrement control, for example for the speed of a motor. Or, you can use a numeric input enable button or numeric input cursor point.

For information about the numeric input enable button, see page 21-30. For information about the numeric input cursor point, see page 21-32.

Setting up momentary push buttons

Use the momentary push button to start a process or action. When pressed, the button changes a tag to one value, and when released, the button changes the tag to another value.

The momentary push button's states can perform one of three kinds of actions:

- Change the Value connection to 1 when the button is pressed, and to 0 when the button is released. This kind of button is called **normally open**, because its released state is off. Pressing the button completes the circuit.
- Change the Value connection to 0 when the button is pressed, and to 1 when the button is released. This kind of button is called **normally closed**, because its released state is on. Pressing the button breaks the circuit.
- Change the Value connection to a **value** you specify. You assign the desired values to the button's press and release actions. For example, 50 when pressed, and 100 when released.

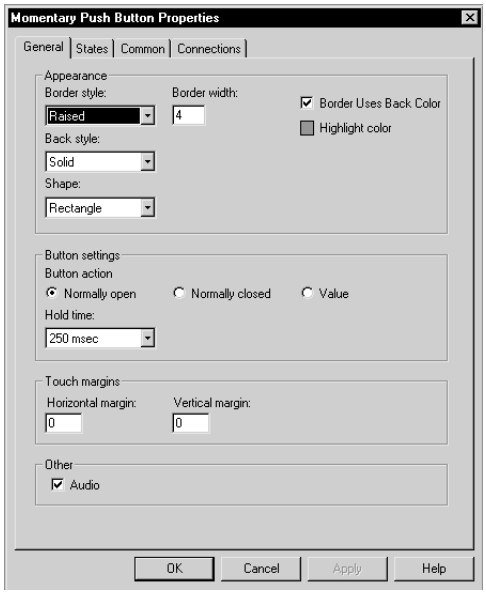
The error state

The button's error state is displayed at run time when:

- the Value connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.

To set up a momentary push button

1. Double-click the momentary push button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the momentary push button looks like at run time, and what type of action the button performs.
   - In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up maintained push buttons

Use the maintained push button to change a setting in a machine or process. Maintained push buttons are not useful for starting or stopping a machine or process.

When pressed the first time, the maintained push button changes a tag to one value. When pressed and released a second time, the button changes the tag to another value.

The error state

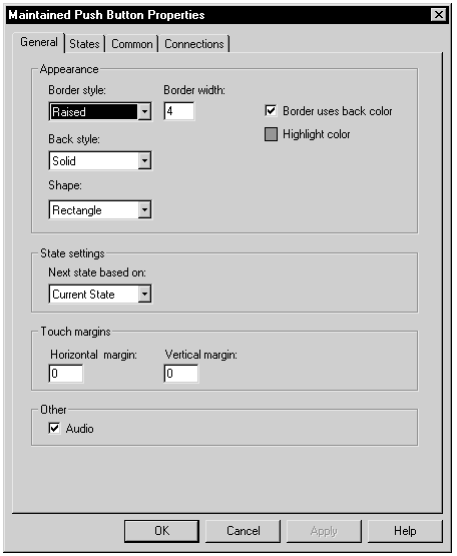
The button's error state is displayed at run time when:

- the Value connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.

The error state is also displayed when the display containing the maintained push button first opens, if the Value connection's value does not match one of the state values you set up.

To set up a maintained push button

1. Double-click the maintained push button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time, and how the button changes state.
   - In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up latched push buttons

The latched push button latches in the On position, and must be unlatched by another button or process to return to the Off position. This type of button is useful for starting a machine or process.

For example, use the latched push button when you want the Value connection to start a process within a programmable controller and remain set until the process is completed. You can also use a latched push button when you have a controller with a long program or long update times.

When the operator presses a latched push button, it changes the Value connection to one value, and remains at that value until the Handshake connection gives the signal to unlatch the button.

The error state

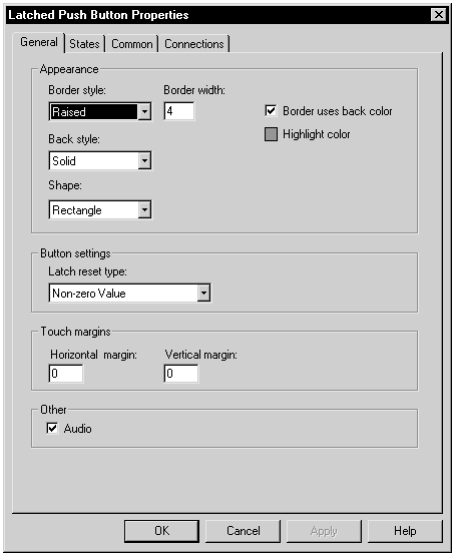
The button's error state is displayed at run time when:

- the Value connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.
- The Handshake connection is unassigned.

The error state is also displayed when the display containing the latched push button first opens, if the Value connection's value does not match one of the state values you set up.

To set up a latched push button

1. Double-click the latched push button.



2. In the button's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the button looks like at run time, and how to unlatch it.
   - In the States tab, specify what the button does when it is latched and unlatched. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up multistate push buttons

The multistate push button displays—and allows an operator to cycle through—multiple options consecutively. The multistate push button displays the current state of a process or operation by showing a different color, caption, or image to reflect different states.

Each time the operator presses the button, the tag changes to the value for the next state. When the button is in its last state and the operator presses the button, the button returns to its first state.

The error state

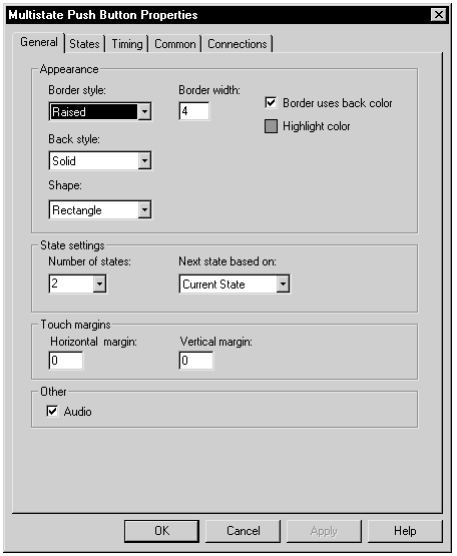
The button's error state is displayed at run time when:

- the Value connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.

The error state is also displayed when the display containing the multistate push button first opens, if the Value connection's value does not match one of the state values you set up.

To set up a multistate push button

1. Double-click the multistate push button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like, and how the button changes state.
   - In the States tab, specify what the button does when it is pressed and released. For tips about setting up states, see page 21-2.
   - In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 21-12.

- In the Common tab, specify the button's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up interlocked push buttons

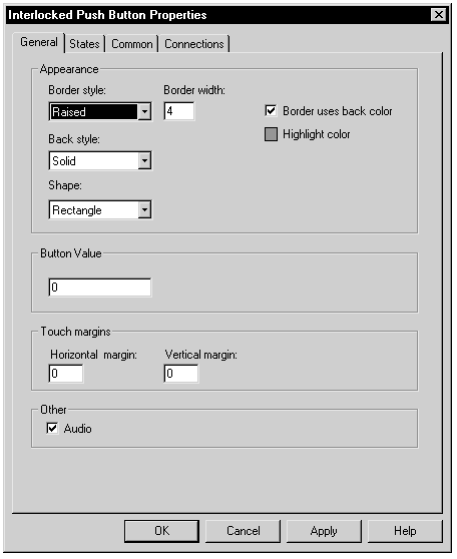
Multiple interlocked push buttons work together and share the same tag. Pressing one button cancels another. Although interlocked push buttons work as a group, you add them to the display one at a time.

When the operator presses one of the interlocked push buttons, the tag assigned to its Value connection changes to one value. When the operator presses a different interlocked push button, the tag changes to another value. Assign the same tag to each button's Value connection.

You can also use a single interlocked push button to send a value to a tag.

To set up an interlocked push button

1. Double-click the interlocked push button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify the button's appearance, and the value it sends to the Value connection.
   - In the States tab, specify what the button looks like when it is pressed and released at run time. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag with which the button exchanges data. Interlocked push buttons have no Indicator connection. For information about assigning tags, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up ramp buttons

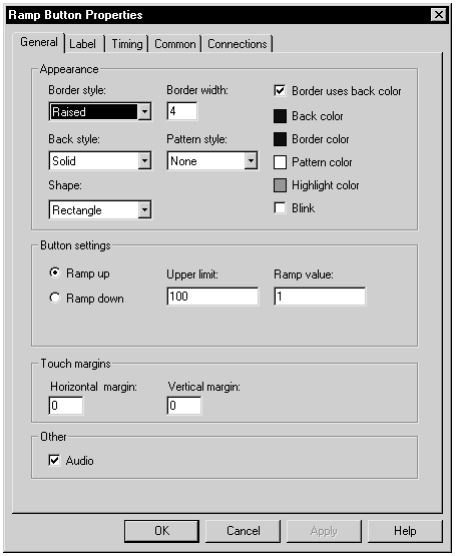
Use the ramp button to increase or decrease the value of a tag.

Ramp buttons can change a tag by either an integer or floating-point value. You can use two ramp buttons together to create an increment/decrement control.

Each time the operator presses the button, the tag value increases or decreases, depending on how you set up the button.

To set up a ramp button

1. Double-click the ramp button.



2. In the button's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the button looks like, and whether the button ramps a value up or down.
   - In the Label tab, specify what text or image to display on the button.
   - In the Timing tab, set up whether or not the button press repeats automatically when the operator presses and holds the button down. You can also set up the rate at which the button press repeats. For more information about auto repeat, see page 21-12.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag with which the button exchanges data. Ramp buttons have no Indicator connection. For information about assigning tags, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up numeric displays

Use the numeric display object to show the operator numeric information from the data source. For example, you might use a numeric display to show the current temperature of an oven.

How values are displayed

The numeric display shows the value of the Value connection at the data source. The value shown depends on whether the Value connection value is a floating-point number or an integer. Integer values are displayed as is. Floating-point values are rounded to fit the display. The decimal places option also affects how floating point numbers are displayed.

For example, if the numeric display is set up to show 6 digits, with one decimal place, 1234.56 is rounded to 1234.6. 1234.44 is rounded to 1234.4. The decimal counts as one of the digits.

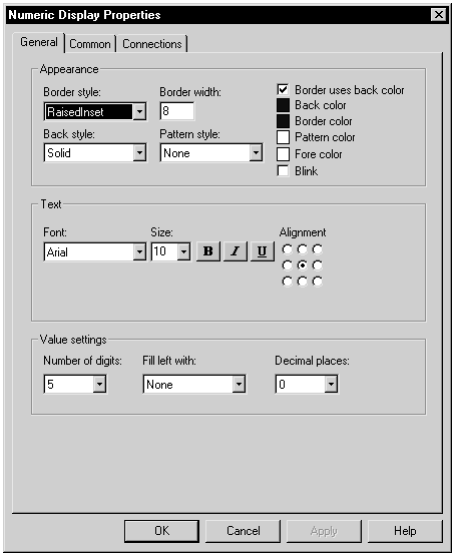
For more information about how values are rounded, see page 7-2.

Problems with displaying values

- If the Value connection is unassigned, the numeric display is filled with asterisks (*).
- If the integer portion of the value, including the decimal point and minus sign, contains more digits than specified for the display, the numeric display is filled with asterisks.
- If the numeric display is sized so that the value cannot be fully displayed, the value is truncated and the last displayable digit is replaced with an asterisk.
- If the value doesn’t fit on the first line of the display, and there is room for a second line, the value continues onto the second line.

To set up a numeric display

1. Double-click the numeric display.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the numeric display looks like at run time.
   - In the Common tab, specify the object's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression whose data is displayed. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up numeric input enable buttons

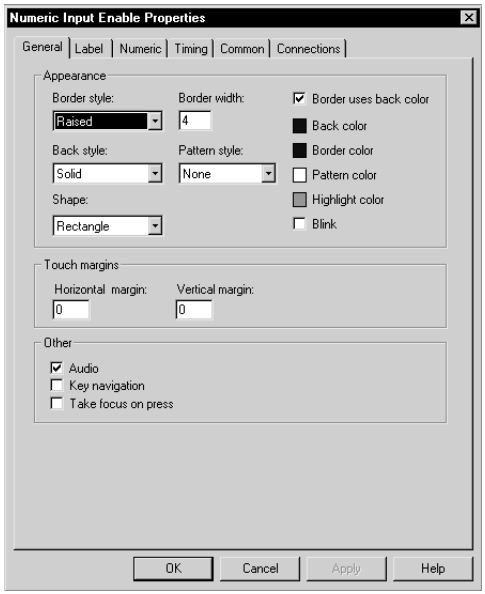
The operator can press the numeric input enable button to open a numeric pop-up keypad or scratchpad. The operator can enter a number in the keypad or scratchpad, and then send the number to the data source.

For more information about using the numeric input enable button at run time, see page 17-6.

You can also set up the numeric input enable button to work as a ramp button. In this case, when the button has focus, the operator can press a move up or move down button to change a tag by either an integer or floating-point value. The operator can also press the Up Arrow or Down Arrow on the keyboard or keypad to ramp the value.

To set up a numeric input enable button

1. Double-click the numeric input enable button.



2. In the button's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the button looks like at run time, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the operator can press the button to give it focus.
   - In the Label tab, specify what text or image to display on the button.
   - In the Numeric tab, set up which pop-up window opens (if any), the ramp value (if any), minimum and maximum values to send to the data source, and decimal point settings.
   - In the Timing tab, set up the timing and handshake settings for the Enter key. These settings do not apply when you ramp a value. For information about using Enter key handshaking, see page 21-13.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expressions with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up numeric input cursor points

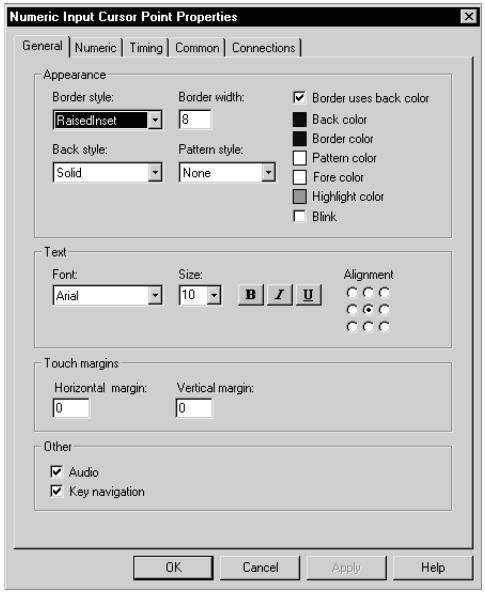
The operator can activate the numeric input cursor point to open a numeric pop-up keypad or scratchpad. The operator can enter a number in the keypad or scratchpad, and then send the number to the data source. The cursor point displays the current value at the data source (if you assign a tag to the Indicator connection).

For more information about using the numeric input cursor point at run time, see page 17-6.

You can also set up the numeric input cursor point to work as a ramp button. In this case, when the cursor point has focus, the operator can press a move up or move down button to change a tag by either an integer or floating-point value. The operator can also press the Up Arrow or Down Arrow on the keyboard or keypad to ramp the value.

To set up a numeric input cursor point

1. Double-click the numeric input cursor point.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the cursor point looks like at run time, and whether the operator can navigate to the cursor point using the keys on the keyboard or keypad.

- In the Numeric tab, set up which pop-up window opens (if any), the ramp value (if any), minimum and maximum values to send to the data source, decimal point settings, and display settings.
- In the Timing tab, set up the timing and handshake settings for the Enter key. These settings do not apply when you ramp values. For information about using Enter key handshaking, see page 21-13.
- In the Common tab, specify the object's spacial properties, name, and visibility.
- In the Connections tab, specify the tags or expression with which the numeric input cursor point exchanges data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up string displays

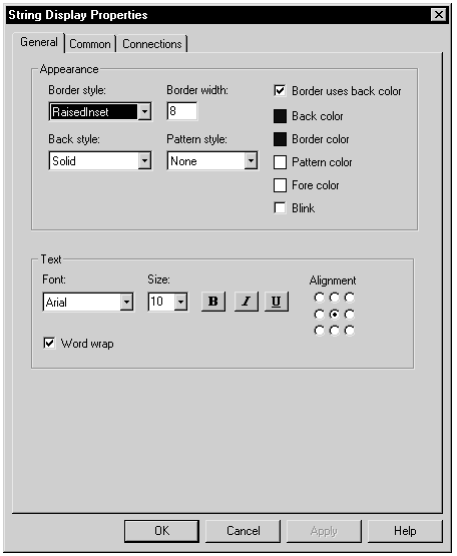
Use the string display object to show the operator messages from the data source. For example, you might set up the data source to generate strings that report on the state of a process or operation, or that provide instructions about what the operator needs to do next. At run time the display shows the operator the string value of the Value connection at the data source.

How values are displayed

- If the Value connection is unassigned, the string display is blank.
- FactoryTalk View displays the number of characters (bytes) specified for the tag assigned to the Value connection.

To set up a string display

1. Double-click the string display.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the string display looks like at run time.
   - In the Common tab, specify the string display's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression whose data is displayed. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

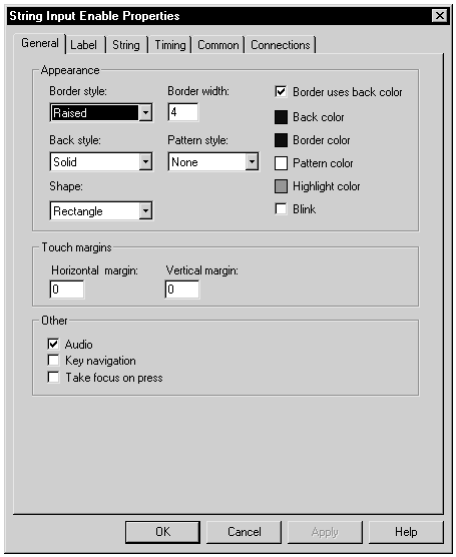
Setting up string input enable buttons

The operator can press the string input enable button to open a string pop-up scratchpad or keyboard. The operator can enter text in the scratchpad or keyboard, and then send the string to the data source.

For more information about using the string input enable button at run time, see page 17-6.

To set up a string input enable button

1. Double-click the string input enable button.



2. In the button's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the button looks like at run time, whether the operator can navigate to the button using the keys on the keyboard or keypad, and whether the operator can press the button to give it focus.
   - In the Label tab, specify what text or image to display on the button.
   - In the String tab, specify the pop-up to open and the number of characters to accept.
   - In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 21-13.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression with which the button exchanges data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

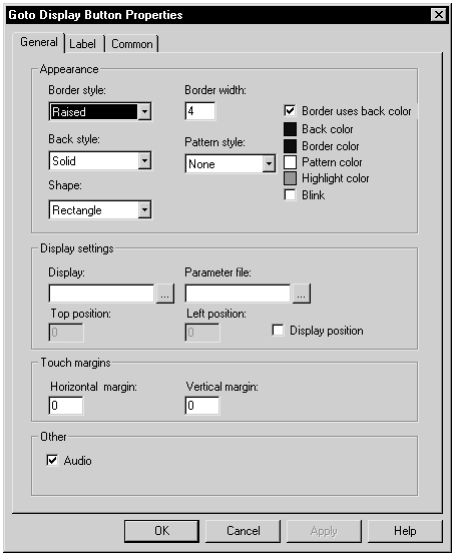
Setting up goto display buttons

Use the goto display button to open a graphic display. You can create as many goto display buttons as you need, but each button opens a single display only.

For more information about using goto display buttons, see page 13-4.

To set up a goto display button

1. Double-click the goto display button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time, the display to open, and the parameter file to use, if any.
   - In the Label tab, specify what text or image to display on the button.
   - In the Common tab, specify the button's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

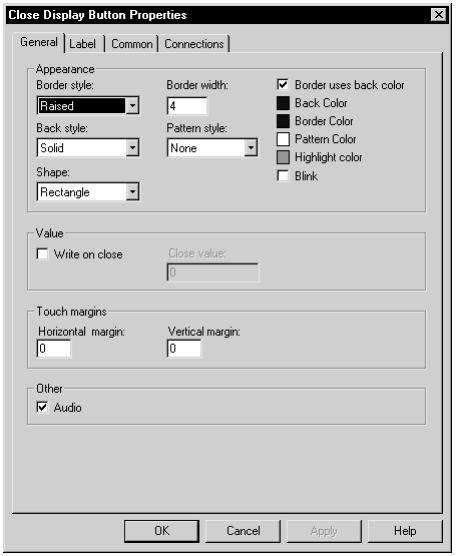
Setting up close display buttons

When the operator presses a close display button at run time, the graphic display that the button is on closes. You can set up the button to write out a value when the display closes.

For more information about using close display buttons, see page 13-6.

To set up a close display button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time, and whether to write out a value when the display closes.
   - In the Label tab, specify what text or image to display on the button.
   - In the Common tab, specify the button's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag with which the button exchanges data. For information about assigning tags, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

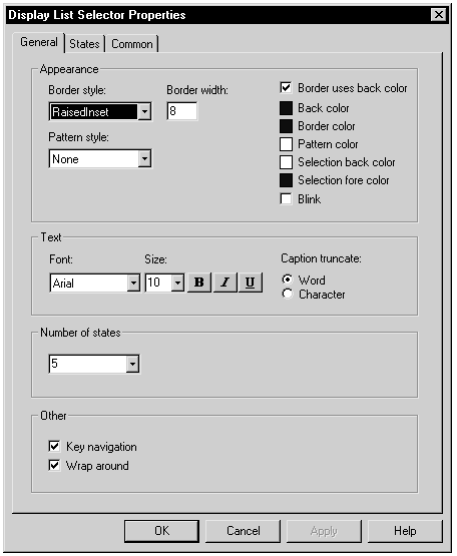
Setting up display list selectors

The display list selector is a list of graphic displays in the application. Each graphic display is represented by a different state in the display list selector. An operator can scroll through the list of displays, and then select the display to open by pressing an enter button or Enter key.

For more information about using display list selectors, see page 13-6.

To set up a display list selector

1. Double-click the display list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the selector looks like at run time, its number of states, whether the operator can navigate to the selector using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
   - In the States tab, specify which graphic display to use for each of the display list selector's states, and how each graphic display is named in the list. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the display list selector's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

How to use indicators

Indicators display the status of processes or operations by showing different colors, captions, or images to reflect different states.

You can create different kinds of indicators to suit your needs:

Multistate indicators display the current state of a process or operation by showing a different color, caption, or image to reflect different states.

Symbols display a symbol that changes color to indicate the state of a process or operation. This allows the operator to see the status of a process or operation at a glance.

List indicators display a list of states for a process or operation, and highlight the current state. Each state is represented by a caption in the list. This type of indicator is useful if you want operators to view the current state, but also see the other possible states. For sequential processes, the list can inform the operator about what happens next.

Setting up multistate indicators

The multistate indicator displays the current state of a process or operation by showing a different color, caption, or image for each state.

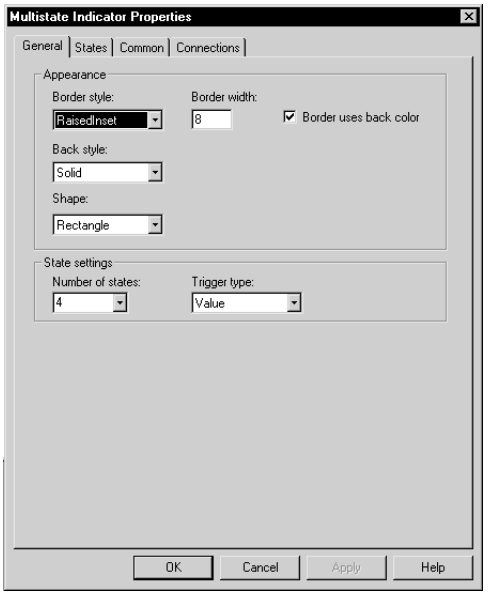
The error state

The multistate indicator's error state is displayed at run time when:

- the Indicator connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.
- the Trigger type is set to Value and the Indicator connection value does not match one of the state values you set up.
- the Trigger type is set to LSB and the position of the least significant bit set in the Value connection does not match one of the state values you set up.
- the Trigger type is set to Value and an array tag has been assigned to the object's Value connection.

To set up a multistate indicator

1. Double-click the multistate indicator.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the indicator looks like, and the number of states for the indicator.
   - In the States tab, specify how the indicator's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the indicator's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression from which the indicator receives data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up symbols

The symbol is an indicator that displays a single monochrome image that changes color to match the state of a process or operation. This allows the operator to see the status of a process or operation at a glance.

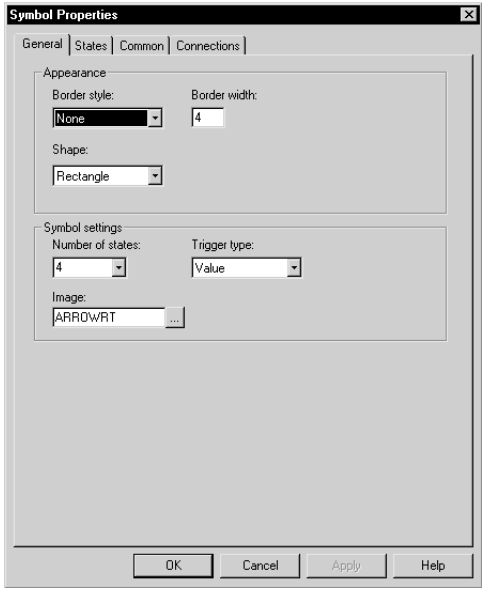
The error state

The symbol's error state is displayed at run time when:

- the Indicator connection is unassigned.
- the Indicator connection's value does not match one of the state values you set up.
- the Trigger type is set to Value and the Indicator connection value does not match one of the state values you set up.
- the Trigger type is set to LSB and the position of the least significant bit set in the Value connection does not match one of the state values you set up.
- the Trigger type is set to Value and an array tag has been assigned to the object's Value connection.

To set up a symbol

1. Double-click the symbol.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the symbol looks like at run time, and the number of states for the symbol.
   - In the States tab, specify how the symbol's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 21-2.

- In the Common tab, specify the symbol's spacial properties, name, and visibility.
- In the Connections tab, specify the tag or expression from which the symbol receives data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up list indicators

The list indicator displays a list of states for a process or operation, and highlights the current state.

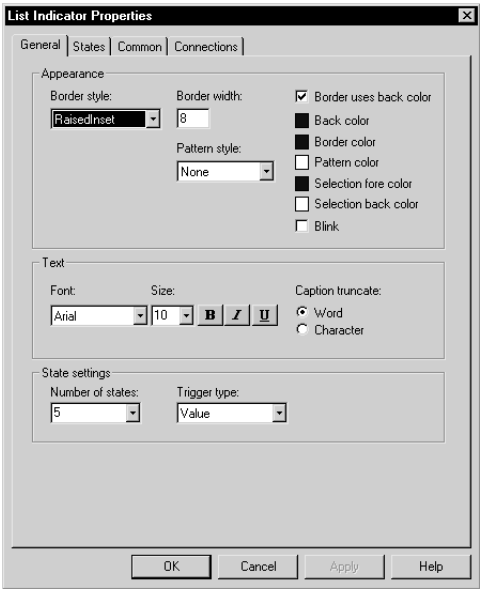
Each state is represented by a caption in the list. This type of indicator is useful if you want operators to view the current state, but also see the other possible states.

For sequential processes, the list can inform the operator about what happens next.

The list indicator has no error state. If the value of the Indicator connection does not match any of the available states, none of the states is highlighted.

To set up a list indicator

1. Double-click the list indicator.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the indicator looks like at run time, and the number of states for the indicator.
   - In the States tab, specify how the indicator's appearance changes when its tag or expression's value changes. For tips about setting up states, see page 21-2.
   - In the Common tab, specify the indicator's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression from which the indicator receives data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

How to use bar graphs, gauges, and scales

Bar graphs and gauges display graphical representations of numeric values. The scale is used with bar graphs to indicate the range of values for the bar graph.

Bar graphs make it easy to compare values

Bar graphs display numeric values in bar graph format. They are useful for allowing comparisons between multiple values, or for representing the fill levels of tanks for which a reading on a vertical scale is appropriate.

For example, one bar graph can show the required level of a tank, and a second bar graph can show the actual level of the tank. The first graph can change to represent the required level for a particular application, and the second graph changes as the actual level in the tank rises or drops.

Bar graphs are more useful than numeric displays when it's important for the operator to analyze the relationships between numeric values. It's easier for the operator to see that one graph is at a lower level than the other, or that one's fill is blue and the other's is yellow, than it is to subtract one numeric value from another.

Thresholds change a bar graph's fill color

As it fills, a bar graph can change its fill color to help an operator recognize abnormal conditions. The change in color happens when the tag (or expression) value crosses a threshold you set up for the graph. For example:

- If the temperature of an oven is lower than required for a recipe, the bar graph can show the temperature in blue.

- If the temperature is in the correct range for the recipe, the bar graph can show the temperature in green.
- If the temperature is higher than the recipe allows, the bar graph can show the temperature in red.

If you want to use a different fill color on a bar graph to show abnormal conditions, make sure enough of the new fill color is visible when the abnormal condition occurs that the operator can see the condition.

Use bar graphs with scales to show limits

Unlike gauges, bar graphs do not have integrated scales. You can show values on a bar graph using a scale graphic object and text.

Scales consist of major ticks, represented by long lines, and minor ticks, represented by short lines. To indicate the values of major or minor ticks, use text objects.

Gauges make it easy to see limits

Gauges display numeric values in dial format. They are useful for displaying a value in relation to its lower and upper limits.

For example, a temperature gauge shows the current temperature in relation to its minimum and maximum extremes. By looking at the position of the needle on the gauge (pointing left, up, or right), the operator can tell at a glance whether the temperature is nearer its lower limit, nearer the middle, or nearer its upper limit.

Gauges are used instead of numeric displays when it's important for the operator to recognize an abnormal condition instantly, either from far away when the scale on the gauge isn't visible, or before the operator has had the opportunity to determine the exact reading on the gauge. This characteristic of gauges is one of the reasons why they are used in automobile instrumentation.

Thresholds change a gauge's fill color

As the needle sweeps higher on a gauge, the gauge can fill the area behind the needle with a color. The gauge can change its fill color to help an operator recognize abnormal conditions. The change in color happens when the tag (or expression) value crosses a threshold you set up for the gauge. For example:

- If the temperature of an oven is lower than required for a recipe, the gauge can show the temperature in blue.
- If the temperature is in the correct range for the recipe, the gauge can show the temperature in green.

- If the temperature is higher than the recipe allows, the gauge can show the temperature in red.

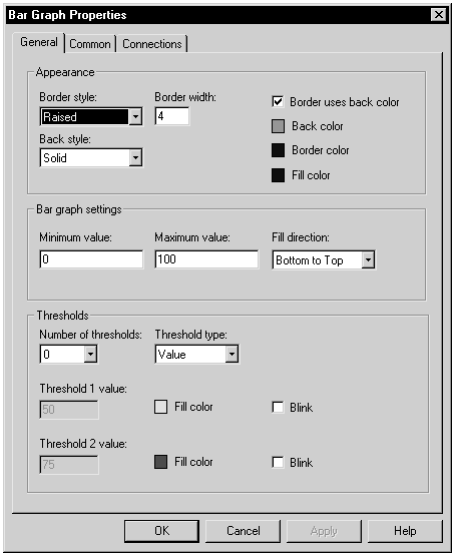
Some people are color blind to red and green, so don't rely on color alone to establish meaning.

Setting up bar graphs

Use bar graphs to represent numeric values by filling and emptying the object as the values rise and fall.

To set up a bar graph

1. Double-click the bar graph.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the bar graph looks like at run time.
   - In the Common tab, specify the bar graph's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression from which the bar graph receives data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

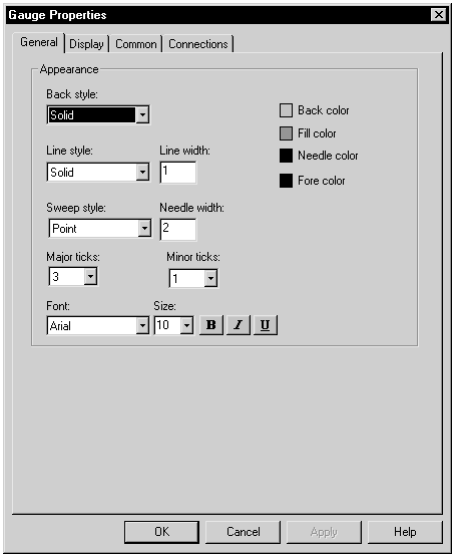
Setting up gauges

The gauge graphic object represents numeric values using a needle on a dial.

At run time, the gauge indicates the value of a tag or expression in relation to the gauge's minimum and maximum values.

To set up a gauge

1. Double-click the gauge.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the gauge looks like at run time.
   - In the Display tab, specify how the gauge displays values at run time.
   - In the Common tab, specify the gauge's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression from which the gauge receives data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

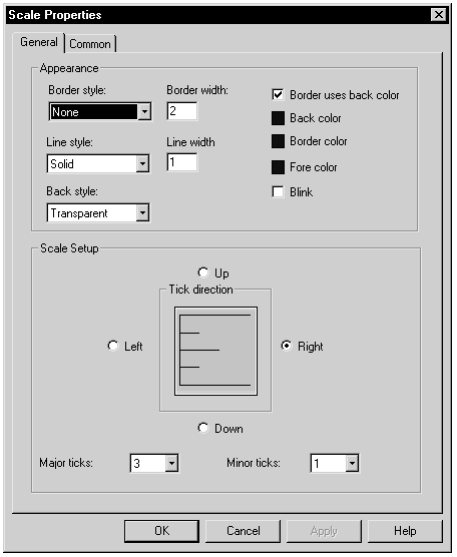
Setting up scales

Use the scale graphic object to show the possible range of values for a bar graph.

To place value labels on the scale (to create a legend), use text objects. Create the text objects and then position them next to the tick marks on the scale. Group all the objects with the bar graph.

To set up a scale

1. Double-click the scale.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the scale looks like at run time.
   - In the Common tab, specify the scale's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up control list selectors

Control list selectors allow an operator to scroll through a list of states for a process and select one of the states. A highlight in the list shows the current state.

A control list selector can show several states at the same time, but only one state can be selected at a time. As the operator scrolls through the list, each successive state is selected

automatically. If you want the operator to confirm the selection of a particular state before the state's value is written to the programmable controller, create an enter button on the same display as the control list selector, and select the "Write on enter" option.

The operator can scroll through the control list selector using:

- key button graphic objects. These are graphic objects that duplicate the functions of keyboard keys.
- the arrow keys and Enter key on a terminal's keypad.
- the arrow keys and Enter key on a keyboard.

Using buttons with the control list selector

The operator presses the buttons to scroll up or down the list, or to make selections from the list. The buttons can be set up to work with the control list selector that has input focus, or you can link the buttons to a specific control list selector.

Use this button	To do this
Backspace	Move the cursor back to the highlighted item in the list.
End	Move to the bottom item in the list.
Enter	Select the item the cursor is pointing to.
Home	Move to the top item in the list.
Move down	Move down one item in the list.
Move up	Move up one item in the list.
Page down	Move down one page in the list.
Page up	Move up one page in the list.

How Enter key handshaking works

When the operator selects a state in the control list selector and presses an enter button or Enter key, the highlighted state's value is written to the programmable controller or device.

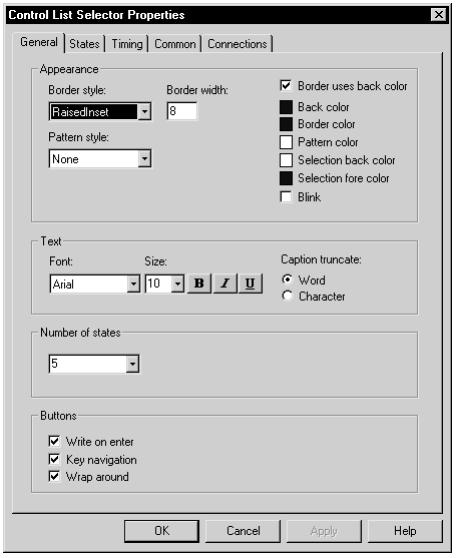
You can use Enter key handshaking to hold the value of the tag at the programmable controller or device for a specific period of time, to ensure the value is read before the control list selector overwrites the value with a new value.

Enter key handshaking only works if the Write on enter check box is selected (in the General tab).

For more information about using Enter key handshaking, see page 21-13.

To set up a control list selector

1. Double-click the control list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the selector looks like at run time, its number of states, whether to write out a value when the operator presses the Enter key, whether the operator can navigate to the selector using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
   - In the States tab, set up the value and caption for each state. The value defines the control list selector's action, by changing the Value connection to the value you specify when the selected state is in effect. For tips about setting up states, see page 21-2.
   - In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 21-13.
   - In the Common tab, specify the control list selector's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expression with which the control list selector exchanges data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up piloted control list selectors

Piloted control list selectors allow an operator or remote device to scroll through a list of states for a process and select one of the states. A highlight in the list shows the current state.

Choosing between piloted control list selectors and control list selectors

Piloted control list selectors include the following features that control list selectors do not:

- The states can be selected either by an operator, or remotely, for example by a programmable controller, or by both an operator and a remote device. Individual states can be turned off, to prevent them from being selected by either the operator or the remote device.

  Control list selectors do not allow states to be turned off.

- The state values of all of the items that are visible in the list can be written to the Visible States connection when the list scrolls. The Visible States connection must be a data-server tag that supports arrays. You cannot use an HMI tag.

  Control list selectors have no Visible States connection.

- The state value of the item at the top of the list can be written to the Top Position connection when the list scrolls.

  Control list selectors have no Top Position connection.

How piloted control list selectors work at run time

A piloted control list selector can show several states at the same time, but only one state can be selected at a time.

You can set up the piloted control list selector to be operator-controlled or remote-controlled by assigning tags or expressions in the Connections tab.

You can also set up individual states to be operator-controlled, remote-controlled, both, or none. If set to none, the state is turned off.

If the piloted control list selector is set up to be operator-controlled, and the operator selects a remote-controlled state, or one that is turned off, a hollow cursor is displayed.



The operator can select this state.



The operator cannot select this state.

Using buttons with the piloted control list selector

When a piloted control list selector is operator-controlled, it works with:

- key button graphic objects. These are graphic objects that duplicate the functions of keyboard keys.
- the arrow keys and Enter key on a terminal's keypad.
- the arrow keys and Enter key on a keyboard.

The operator presses the buttons to scroll up or down the list, or to make selections from the list. The buttons can be set up to work with the piloted control list selector that has focus, or with a specific piloted control list selector. You can use the buttons listed in the table on page 21-48.

Selecting items in the list

Each state can be selected directly by an operator, or remotely by a device such as a programmable controller. The operator or the controller scrolls through the list to select a different state or a different group of visible states.

When an operator or remote device selects a state, the value assigned to the selected state is written to the piloted control list selector's Value connection. If the state is turned off, the state's value is not written to the Value connection.

If the operator attempts to select a state that is remote-controlled, the state's value is not written to the Value connection.

Scrolling through the list

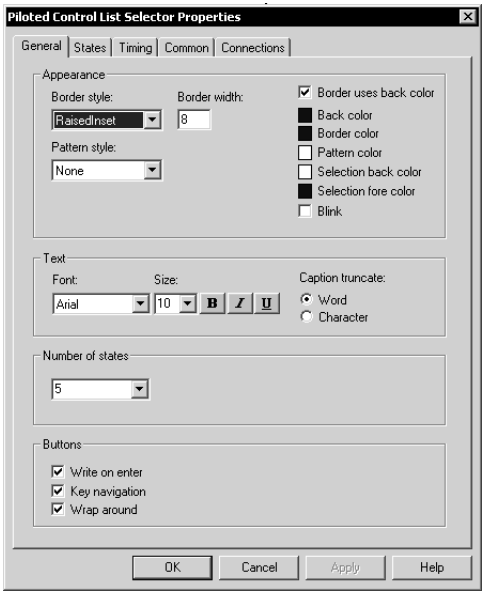
If the piloted control list selector contains more states than can be displayed in the list simultaneously, the value of the Top Position connection (if assigned) changes whenever the item at the top of the list changes.

If the Visible States connection is assigned, the values assigned to all visible states are written to the Visible States connection whenever the list scrolls. To use this feature, the assigned tag must support arrays, and the array must be the same length as the number of

visible states in the piloted control list selector. For information about using array tags, see page 9-15.

To set up a piloted control list selector

1. Double-click the piloted control list selector.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the selector looks like at run time, its number of states, whether to write out a value when the operator presses the Enter key, whether the operator can navigate to the selector using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
   - In the States tab, set up the value and caption for each state. Also specify whether each state can be selected, and by whom. For tips about setting up states, see page 21-2.
   - In the Timing tab, set up the timing and handshake settings for the Enter key. For information about using Enter key handshaking, see page 21-13.
   - In the Common tab, specify the piloted control list selector's spacial properties, name, and visibility.

- In the Connections tab, specify the tags or expression with which the piloted control list selector exchanges data. For information about assigning tags and expressions, see page 20-36.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up local message displays

Use local message displays to provide the operator with information about what to do next, or with information about a process. At run time, the local message display shows one message at a time.

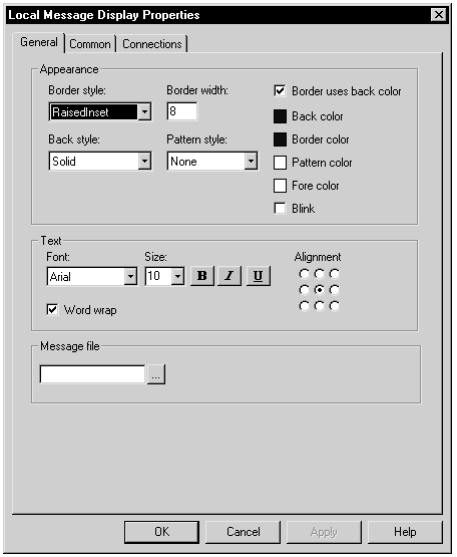
To use local messages, create a local message display object in a graphic display, and then assign a local message file to the local message display object.

For more information about local messages, see page 19-25.

If no message exists for the trigger value that matches the value of the Value connection, the display is filled with question marks (?).

To set up a local message display

1. Double-click the local message display.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the local message display looks like at run time and which local message file to use.
   - In the Common tab, specify the local message display's spacial properties, name, and visibility.
   - In the Connections tab, specify the tag or expression with which the local message display exchanges data. For information about assigning tags and expressions, see page 20-36.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

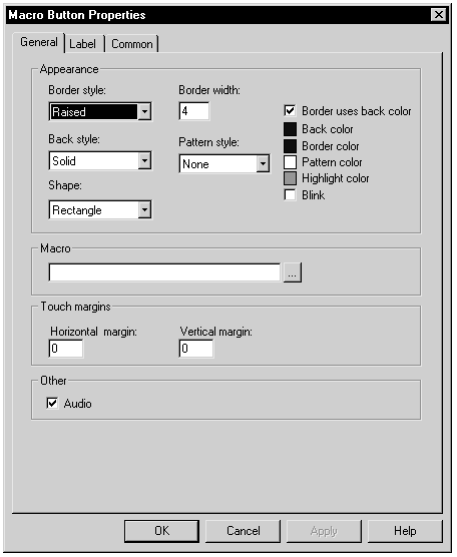
Setting up macro buttons

Macro buttons run macro files that assign values to tags when the button is pressed.

For information about creating macros, see Chapter 30.

To set up a macro button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time, and the macro to run when the button is pressed.
   - In the Label tab, specify what text or image to display on the button.
   - In the Common tab, specify the button's spacial properties, name, and visibility.

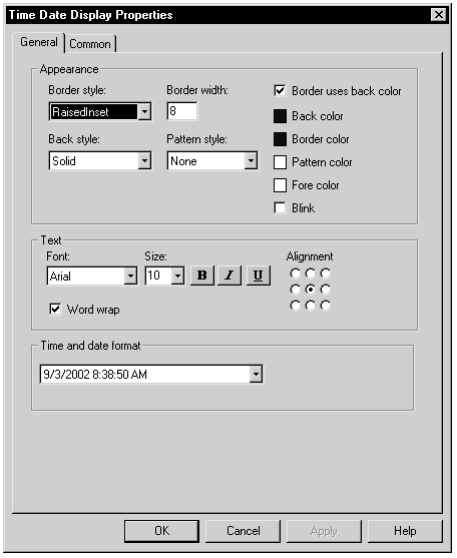
   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up time and date displays

A time and date display shows the current time and date in a graphic display.

To set up a time and date display

1. Double-click the time and date display.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the time and date display looks like at run time.
   - In the Common tab, specify the time and date display's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

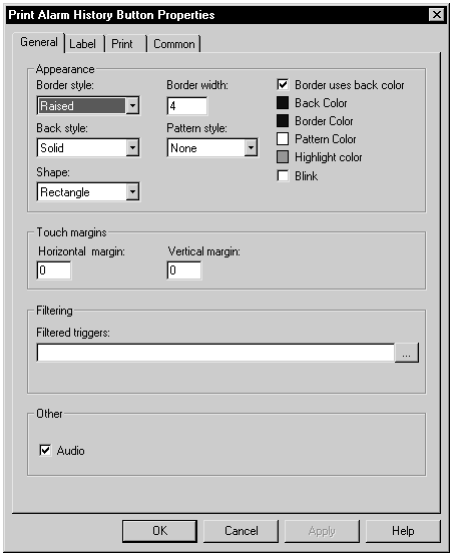
Setting up print alarm history buttons

The operator can press the print alarm history button to print a report of all the alarm messages in the alarm log file. The report can include the time alarms occurred and were acknowledged.

For information about the alarm log file, see page 9-10.

To set up a print alarm history button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time and whether to filter the alarms to include in the report.
   - In the Label tab, specify what text or image to display on the button.
   - In the Print tab, specify what information to print on the report.
   - In the Common tab, specify the button's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

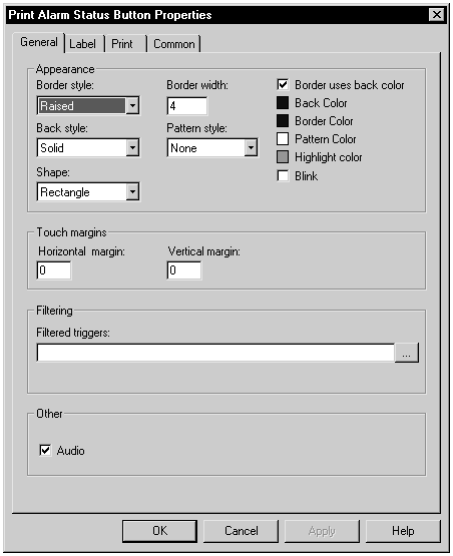
3. When you are finished, click OK.

Setting up print alarm status buttons

The operator can press the print alarm status button to print a report of the status of alarms that have been defined for the application (in the Alarm Setup editor). The report can include all alarms, active alarms only, or only alarms that have been active since the alarm status was last reset. The report can include how many times each alarm was triggered, and the accumulated time in alarm.

To set up a print alarm status button

1. Double-click the button.



2. In the button's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the button looks like at run time and whether to filter the alarms to include in the report.
   - In the Label tab, specify what text or image to display on the button.
   - In the Print tab, specify what information to print on the report.
   - In the Common tab, specify the button's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

Setting up alarm lists

The alarm list graphic object displays a list of alarms from the alarm log file. It can show all the alarms in the alarm log file, or any combination of active, inactive, acknowledged, and unacknowledged alarms. It can also show the alarms for specific alarm triggers.

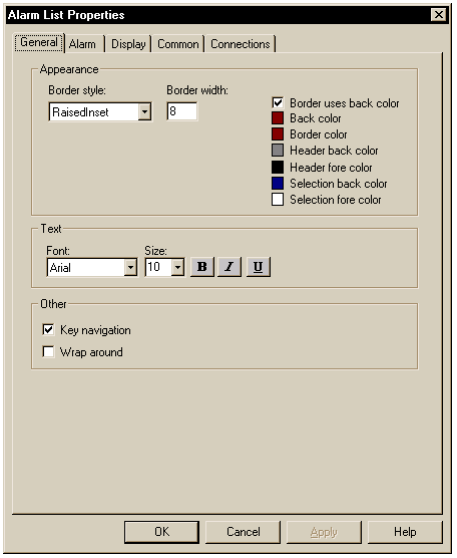
You can assign a tag or expression to the ActiveAcknowledged, ActiveUnacknowledged, InactiveAcknowledged, or InactiveUnacknowledged connections, to allow the data source to control the type of alarms to display in the list. When one of the connection's values is a non-zero value, the related type of alarm is displayed in the alarm list. The connection value overrides the setting for the list in the Alarm List Properties dialog box.

FactoryTalk View comes with two graphic displays, [ALARM MULTI-LINE] and [HISTORY], that contain alarm list objects. The [HISTORY] display comes in two sizes. These displays are in the graphic library. You can copy the library displays into your application and use the displays and objects as is, customize them, or create your own.

For more information about using alarm lists, see page 9-32. For information about the alarm log file, see page 9-10.

To set up an alarm list

1. Double-click the alarm list.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the alarm list looks like at run time, whether the operator can navigate to the list using a keyboard, and how scrolling works for the list.
   - In the Alarm tab, specify the columns to include in the list, the number of lines per alarm, and the triggers to filter by, if any.
   - In the Display tab, specify the types of alarms to display in the list, and the appearance of the different alarm conditions.
   - In the Common tab, specify the alarm list's spacial properties, name, and visibility.
   - In the Connections tab, specify the tags or expressions to use to control the type of alarms to display in the list. The use of these connections is optional.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

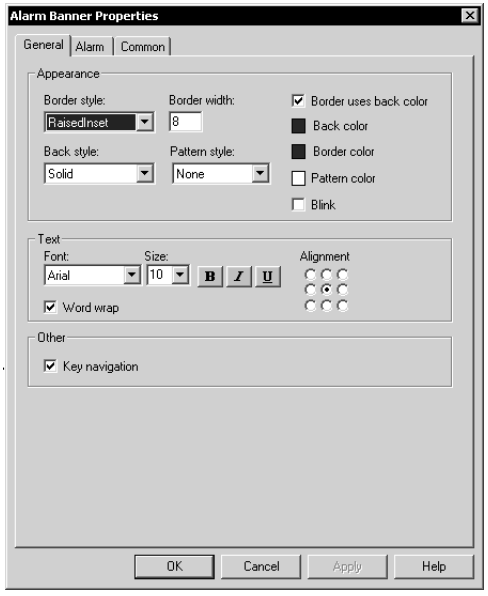
Setting up alarm banners

The alarm banner graphic object displays a single unacknowledged alarm message. For example, the banner could display a message that warns the operator that the pressure in a boiler is too high.

The [ALARM] display and the [ALARM BANNER] graphic library both contain an alarm banner object. You can use the displays and objects as is, customize them, or create your own. For more information about using alarm banners, see page 9-33.

To set up an alarm banner

1. Double-click the alarm banner.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the banner looks like at run time and whether the operator can navigate to the banner using the keys on the keyboard or keypad.
   - In the Alarm tab, set up whether to queue alarms, whether to display all alarms or active alarms only, and which alarms to include in a filtered trigger list.
   - In the Common tab, specify the alarm banner's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up alarm status lists

The alarm status list graphic object displays the status of alarms, including how many times an alarm has been triggered and how long it has been active. For example, the list could display an alarm that has been triggered 5 times for a total accumulated time in alarm of 10 minutes.

The operator can view the status list in three different modes:

- All alarms—displays the status of all alarms that have been defined in the Alarm Setup editor.

- Active alarms—displays the status of all the alarms that are currently active.
- Past alarms—displays the status of all the alarms that have been active since the alarm status has been reset.

The operator can press the alarm status mode button to cycle through the different modes.

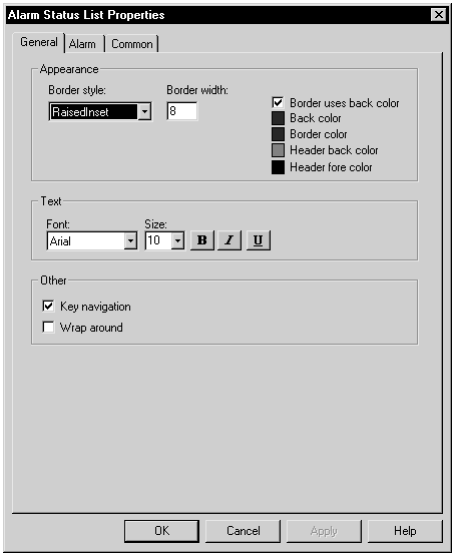
If desired, you can set up the alarm status list to show alarms for specific alarm triggers only.

FactoryTalk View comes with a library graphic display called [STATUS] that contains an alarm status list object, alarm status mode button, and other buttons for working with the list. You can use the display and objects as is, customize them, or create your own.

For more information about using alarm status lists, see page 9-34.

To set up an alarm status list

1. Double-click the alarm status list.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the alarm status list looks like at run time, whether the operator can navigate to the list using a keyboard, and how scrolling works for the list.
   - In the Alarm tab, specify the columns to include in the list, the number of lines per alarm, and the triggers to filter by, if any.

- In the Common tab, specify the alarm status list's spacial properties, name, and visibility.

For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up diagnostics lists

The diagnostics list graphic object displays runtime information about system activity:

- Information messages, such as messages about tag assignments, tag read and write activity, and macro usage.

  These are system-generated information messages. They are not the same messages as the information messages you create and display in the information message display object.

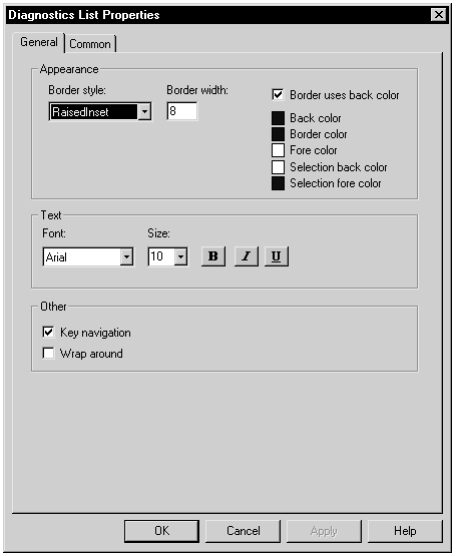
- Warning messages.
- Error messages, such as messages about communication errors, invalid values, and unassigned connections.
- Audit messages about tag writes and tag write failures.

FactoryTalk View comes with a default graphic display called [DIAGNOSTICS] that contains a diagnostics list object. You can use the display and object as is, customize them, or create your own.

For more information about using diagnostics lists, see page 10-11.

To set up a diagnostics list

1. Double-click the diagnostics list.



2. In the object's Properties dialog box, fill in the options on each tab:

   - In the General tab, specify what the list looks like at run time, whether the operator can navigate to the list using the keys on the keyboard or keypad, and whether the cursor wraps from the bottom of the list back to the top.
   - In the Common tab, specify the diagnostics list's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.

3. When you are finished, click OK.

Setting up information message displays

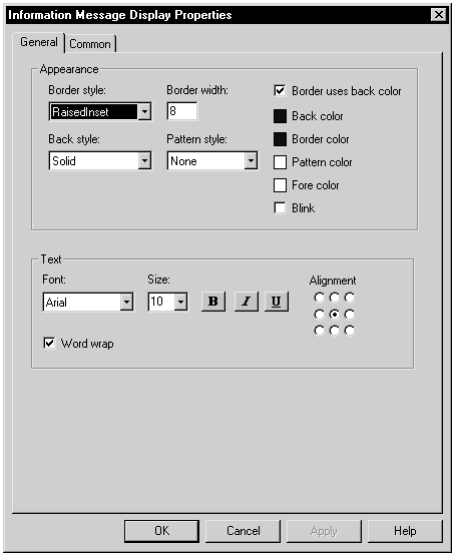
The information message display graphic object gives the operator runtime messages about information that requires immediate attention.

FactoryTalk View comes with a default graphic display called [INFORMATION] that contains an information message display object. You can use the display and object as is, customize them, or create your own.

For more information about using information message displays, see page 27-7.

To set up an information message display

1. Double-click the information message display.



2. In the object's Properties dialog box, fill in the options on each tab:
   - In the General tab, specify what the information message display looks like at run time.
   - In the Common tab, specify the information message display's spacial properties, name, and visibility.

   For details about the options in the tabs, see Help.
3. When you are finished, click OK.

22 Animating graphic objects

This chapter describes:

- the types of animation and which objects support which types of animation.
- using the Animation dialog box.
- testing animation.
- using tag names, tag placeholders, and expressions when attaching animation.
- setting the minimum and maximum values for animation that uses a range of motion.
- using Object Smart Path to define an object's range of motion.
- setting up each type of animation.
- applying animation to groups.
- checking what kind of animation is attached to an object.
- copying and pasting animation.
- setting up animation for global objects.

Types of animation

Animation associates graphic objects with tags so the appearance or position of an object changes to reflect changes to the tag's value. For example, an object's color could change from yellow to orange to red as the tag's value increases. Or a slider could move from left to right as a tag's value increases.

You can use these types of animation:

- color
- fill
- height
- horizontal position
- horizontal slider
- rotation
- vertical position
- vertical slider
- visibility

- width

The Objects 5 Screen Demo sample application contains many examples of animation. In particular, see the graphic displays called "Animation I" and "Animation II."

Which objects can have which types of animation?

This table summarizes which types of objects support which types of animation. For information about the different types of graphic objects, see page 20-1.

These objects	Support these types of animation
Drawing objects, except images, panels, and rounded rectangles	All types
Rounded rectangles	All types except rotation
All other objects	Visibility

You can also attach animation to groups of drawing objects. For more information, see page 22-14.

You can attach as many types of animation to a drawing object (except images and panels) as you like. For example, apply width, height, horizontal position, and vertical position animation to an object to give it the appearance of moving into or out of the display as it shrinks and grows.

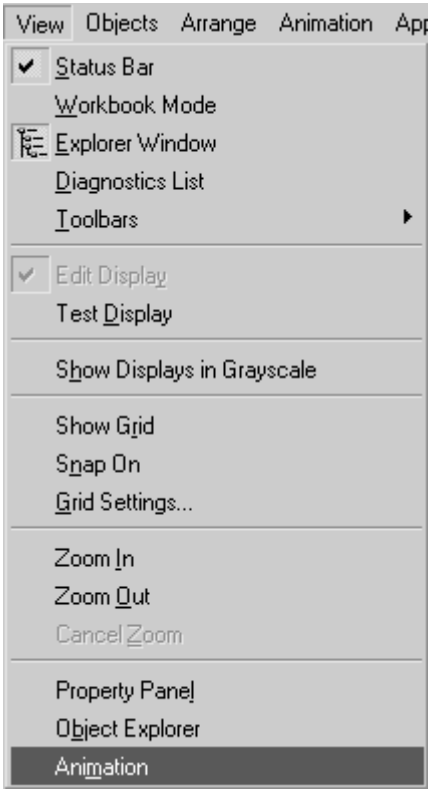
Using the Animation dialog box

To attach animation, use the Animation dialog box.

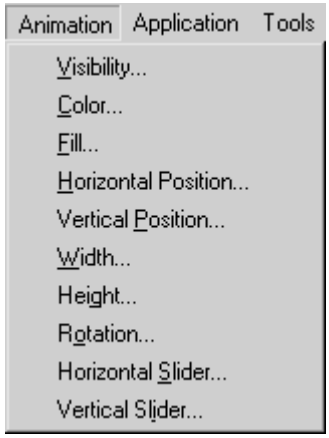
To open the Animation dialog box, do one of the following

- Select an object, and then on the View menu click Animation.
- Select an object, and then on the Animation menu click an animation type. Animation types that are not supported for the selected object are unavailable.
- Right-click an object, select Animation, and then click an animation type. Animation types that are not supported for the selected object are unavailable.

View menu

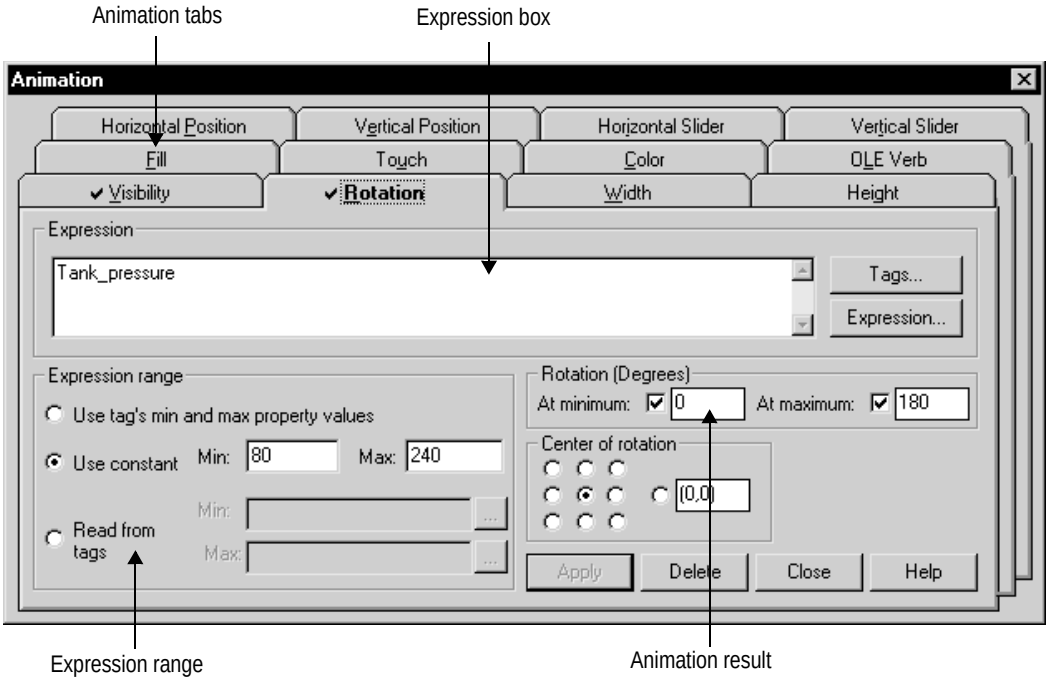


Animation menu



About the Animation dialog box

The Animation dialog box is a floating dialog box, which means you can keep it open all the time and move it around the screen. While it's open you can select other objects and open other dialog boxes.



For details about the parts of the Animation dialog box, see Help.

Using Object Smart Path to visually set animation

Because the Animation dialog box stays open, you can go back and forth between the dialog box and the graphic display. This makes it easy to set the range of motion for an object because you do not have to know how many pixels you want an object to move. Instead, you can set the range of motion visually using the Object Smart Path feature. For details, see page 22-7.

Testing animation



Test Display tool



Edit Display tool

To test the animation you have set up in a graphic display, use the Test Display tool to switch to test mode. When you are finished testing, switch back to edit mode to continue editing.

To switch between test and edit modes

1. On the View menu, click Test Display or Edit Display, or click the Test Display and Edit Display tools.



Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the Display Settings dialog box.

Using tag names and tag placeholders

When setting up animation for objects, you are linking objects to tags. You can specify a tag name or use tag placeholder.

You can use HMI tags or data server tags that already exist, or you can use a new tag name.

Tag placeholders allow you to create displays that can be used with different tags.

You can use tag placeholders in:

- the graphic display that opens when the application is first run.
- graphic displays that are opened using a goto display button.
- graphic displays that are opened using a display list selector.

Use parameter files to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters. For more information about using parameter files and global object parameters, see Chapter 25.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

To create a tag placeholder

1. In the Expression box, type the cross-hatch character followed by a number (no space in between). For example, #1.

Using expressions

Many types of animation can be achieved using expressions. You can use expressions containing tag values, constants, mathematical equations, security functions, and if-then-else logic. A tag name or tag placeholder can be included as part of an expression, or it can stand alone as the entire expression.

For more information about expressions, see Chapter 23.

Setting minimum and maximum values

Many types of animation require a minimum and maximum range for the tag or expression. These values determine the start and end points for a range of motion.

For example, if you specify a minimum of 0 and a maximum of 100, the object will not react to values outside of this range. So, even if the expression has a value of 200, the object does not change from its "At maximum" position.

When setting up animation, select one of these methods for calculating the minimum and maximum values:

- Use tag's min and max property values—select this method to use the minimum and maximum values of the first HMI tag in the expression. If more than one HMI tag is used in the expression, the first HMI tag's minimum and maximum values are used.

  For analog HMI tags, the values are taken from the Minimum and Maximum boxes in the Tags editor. For digital tags, the minimum is 0 and the maximum is 1.

- Use constant—select this method to use numeric constants. Type the minimum and maximum values in the boxes.



Browse button

- Read from tags—select this method to read two tags' values to determine the minimum and maximum values. Type the tag names in the boxes, or click the Browse buttons to open the Tag Browser and select the tags.

  If you use this method, the tags are read when the graphic display opens. Their values at that time are used for the minimum and maximum values. The tags are not read again after this.

Defining a range of motion

To define a range of motion for an object, do one of the following:

- Use the mouse to move the object in the display. This uses the Object Smart Path feature to visually define the range of motion.
- Type values in the At minimum and At maximum boxes.

Motion can be defined in pixels, percentages, or degrees.

Animation that does not use a range of motion

Visibility and color animation do not use a range of motion, because these types of animation represent a change of state, not a range of values.

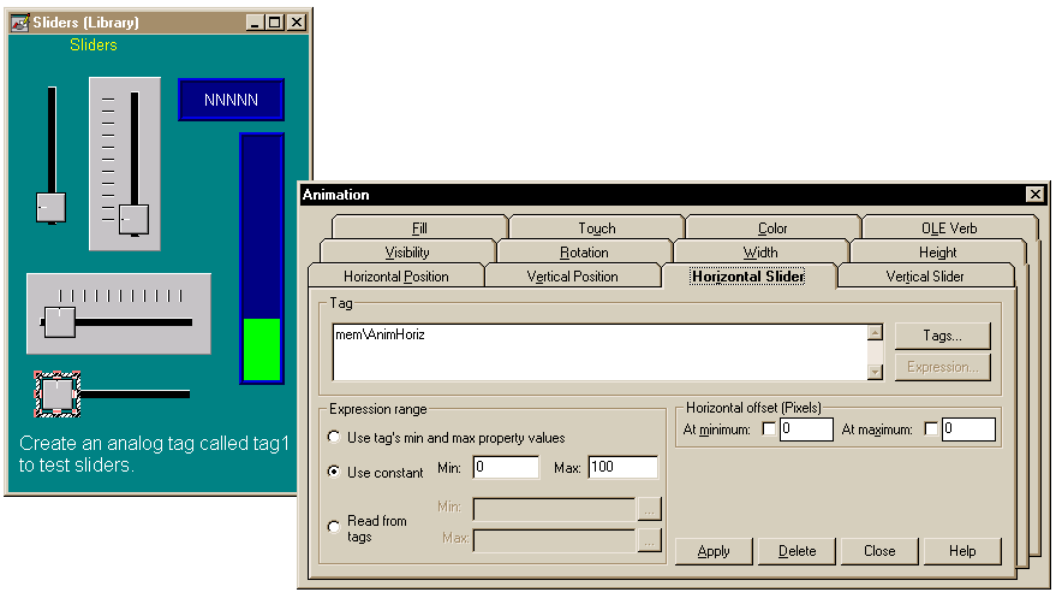
Using Object Smart Path

With Object Smart Path, you can easily set the range of motion for an object. The following example shows how Object Smart Path works.

Example: Using Object Smart Path to define the range of motion for horizontal slider animation

To define a range of motion for a slider object

1. In the Graphics editor, create a slider object using a line and a rectangle, or copy a slider object from the Sliders graphic library.
2. Open the Animation dialog box and click the Horizontal Slider tab.
3. In the display, select the rectangle in the slider object.
4. In the Tag box of the Animation dialog box, specify a tag name.



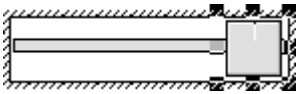
5. In the display, drag the rectangle to the position that will indicate the lowest number in the range.



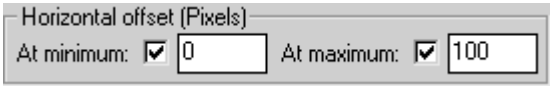
6. In the Animation dialog box, set this position by clicking the At minimum check box.



7. In the display, drag the rectangle to the position that will indicate the highest number in the range.



8. In the Animation dialog box, set this position by clicking the At maximum check box.



9. To save the settings, click Apply.

When you finish setting up the animation, the rectangle returns to its original position.

Setting up the different types of animation

This section describes the different types of animation and provides tips and examples for setting up animation.

Setting up visibility animation

With visibility animation, an object becomes visible or invisible based on a tag value or the result of an expression.

If an object is invisible, it is inactive.

Visibility animation is available for all objects. Visibility animation overrides an object's Visible property.

If you use a tag's value to control visibility animation as well as in an expression to control some other aspect of animation, when you set up visibility, set the Expression true state to "Invisible." If you do not do this, the object could appear briefly in its design-time location and orientation before animating properly.

For more information about setting up visibility animation, see Help.

Example: Using visibility animation to set up security for a graphic display

This example shows how to use visibility animation to control what operators can see. In a graphic display that all users have access to, only the Admin user can see the graphic object to which this animation is attached.

This example uses the security function CurrentUserName( ). The function returns the string value of the Account ID (user name) for the user who is currently logged in.

The CurrentUserName( ) function is case sensitive. All user names use uppercase letters, so make sure that you use uppercase letters in your expression.

To specify which user can view an object in a display

1. Select the object to limit visibility to.
2. Open the Animation dialog box and click the Visibility tab.
3. In the Expression box, type this:

   CurrentUserName( ) == "ADMIN"
4. For the Expression true state, click Visible.
5. Click Apply, and then click Close.

At run time, the object is visible only if the Admin user is logged in.

Setting up color animation

With color animation, an object changes color based on a tag value or the result of an expression. You can specify up to 16 color changes (A to P) for any object. Colors can be solid or blinking. For each color change, specify the value or threshold at which the color is to change and specify the colors to change to. At run time, when the value reaches or crosses the threshold, the color changes.

Color animation is available for all drawing objects except images and panels.

For more information about setting up color animation, see Help.

Example 1: Creating a text object that blinks

This example describes how to create a text object that constantly blinks between two colors. Since the blinking is not based on changes in tag values, the expression is simply a constant value that matches the value for the selected threshold.

For details about creating text, see page 20-13.

1. Select the text object.
2. Open the Animation dialog box, and then click the Color tab.
3. In the Expression box, type 0.

   Zero is the default value for threshold A.
4. In the list box, click threshold A. (Leave the value in the Value box as 0.)
5. For the foreground color, click Blink. (If desired, click Blink for the background color too.)
6. For each color, click the color box, and then click the color to use.
7. Click Apply.

Example 2: Creating an object that changes color as the fill level changes

This example describes how to create a rectangle object that changes color as the object's fill level increases. This example uses a tag called Hopper1\FlourLevel. The tag has a range of 1 to 100. When the flour level reaches 80, the rectangle blinks between gray and yellow to warn the operator that the hopper is nearly full. When the flour level reaches 95, the rectangle blinks between gray and red.

You could use a bar graph object (without animation) to achieve a similar result.

1. Double-click the rectangle to open the Polygon Properties dialog box. Assign these properties to the rectangle:
   - In the Back style box, select Solid.
   - For the Fore color and Back color, select gray.

2. Click OK to close the Polygon Properties dialog box.
3. With the rectangle selected, open the Animation dialog box, and then click the Fill tab.
4. Attach fill animation as follows:
   - In the Expression box, type Hopper1\FlourLevel (this is the tag that monitors the fill level).
   - For Fill Direction, click Up.
5. Click the Color tab, and then attach color animation as follows:
   - In the Expression box, type Hopper1\FlourLevel (the same tag that was used in the Fill tab).

   Set up the color for the normal state
   - In the list box, click A. (In the Value box, leave the value as 0.)
   - For foreground and background colors, click Solid.
   - For each, click the color box, and then click gray (the same gray used for the rectangle).

   Set up the color for the first warning
   - In the list box, click B.
   - In the Value box, type 80.
   - For foreground and background colors, click Blink.



   - For the foreground colors, select gray for the first color and yellow for the second color. Repeat for the background colors.

   Set up the color for the second warning
   - In the list box, click C.
   - In the Value box, type 95.
   - For the foreground and background colors, click Blink.



- For the foreground colors, select gray for the first color and red for the second color. Repeat for the background colors.

6. Click Apply.

Setting up fill animation

With fill animation, the level of fill in an object is based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be half full.

Fill animation is available for all drawing objects (including group objects) except images and panels.

If you select the Inside Only check box, fill animation does not affect objects' borders, line objects, or objects with transparent backgrounds.

For more information about setting up fill animation, see Help.

Setting up horizontal position animation

With horizontal position animation, an object moves horizontally based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be halfway between its minimum and maximum pixel offset.

Horizontal position animation is available for all drawing objects except images and panels.

For more information about setting up horizontal position animation, see Help.

Setting up vertical position animation

With vertical position animation, an object moves vertically based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For

example, if the value of the expression is halfway between the minimum and maximum values, the object will be halfway between its minimum and maximum offset.

Vertical position animation is available for all drawing objects except images and panels.

For more information about setting up vertical position animation, see Help.

Setting up width animation

With width animation, an object's width changes based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object's width will be halfway between its minimum and maximum percentage.

Width animation is available for all drawing objects except images and panels.

For more information about setting up width animation, see Help.

Setting up height animation

With height animation, an object's height changes based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object's height will be halfway between its minimum and maximum percentage.

Height animation is available for all drawing objects except images and panels.

For more information about setting up height animation, see Help.

Setting up rotation animation

With rotation animation, an object rotates around an anchor point based on a tag value (or the result of an expression) in relation to the specified minimum and maximum values. For example, if the value of the expression is halfway between the minimum and maximum values, the object will rotate half the specified amount.

Rotation animation is available for all drawing objects except images, panels, and rounded rectangles. If you apply rotation animation to text, the text rotates around the anchor point but remains in the upright position.

For more information about setting up rotation animation, see Help.

Setting up horizontal slider animation

With horizontal slider animation, you can use a drawing object to set the value of a tag. To do this, define a path for the object. At run time, when the operator moves the object horizontally (using a mouse), the pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider changes as well.

An object can have both vertical and horizontal slider animation.

Horizontal slider animation is available for all drawing objects except images and panels.

For more information about setting up horizontal slider animation, see Help.

Tips

Here are some tips for creating slider objects:

- The Sliders graphic library contains several slider objects you can drag and drop into your graphic displays. Attach slider animation to the button portion of the slider object.
- If you create your own slider object, it's useful to draw an object (for example, a line) to represent the path the slider will move along.

Setting up vertical slider animation

With vertical slider animation, you can use a drawing object to set the value of a tag. To do this, define a path for the object. At run time, when the operator moves the object vertically (using a mouse), the pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider changes as well.

An object can have both vertical and horizontal slider animation.

Vertical slider animation is available for all drawing objects except images and panels.

For tips about creating slider objects, see the previous section.

For more information about setting up vertical slider animation, see Help.

Applying animation to groups

You can apply animation to objects and then group those objects and apply animation to the group. When the display is running, animation is applied as follows:

These types of animation	Are applied like this
Color	Animation attached to individual objects within the group overrides group animation.
Fill	Animation results for both the individual objects and the group are applied.
Horizontal slider, vertical slider	Group animation overrides animation attached to individual objects within the group.

These types of animation	Are applied like this
Height, width, horizontal position, vertical position, and rotation	Animation results for individual objects and the group are combined. For example, if an individual's horizontal position animation result is to offset the object by 100 pixels, and the group's result is to offset the group by 200 pixels, the individual object is offset by 300 pixels.
Visibility	When the group's animation visibility is False (the group is not visible), then no objects in the group are visible, regardless of the animation status of the individual objects. When the group's animation visibility is True (the group is visible), the visibility of an object within the group is determined by the individual object animation.

Test your animation to ensure you achieve the intended results.

To apply animation to objects within groups, use the group edit feature. For details, see page 20-48.

Checking the animation on objects

To see what type of animation has been set up for an object or group of objects, use the Object Explorer, the Animation menu, or the Animation dialog box.

For information about using the Object Explorer to highlight objects that have animation attached, see page 20-25.

To see what type of animation has been set up for objects within a group, use the group edit feature or the Object Explorer. For information about using the group edit feature, see page 20-48.

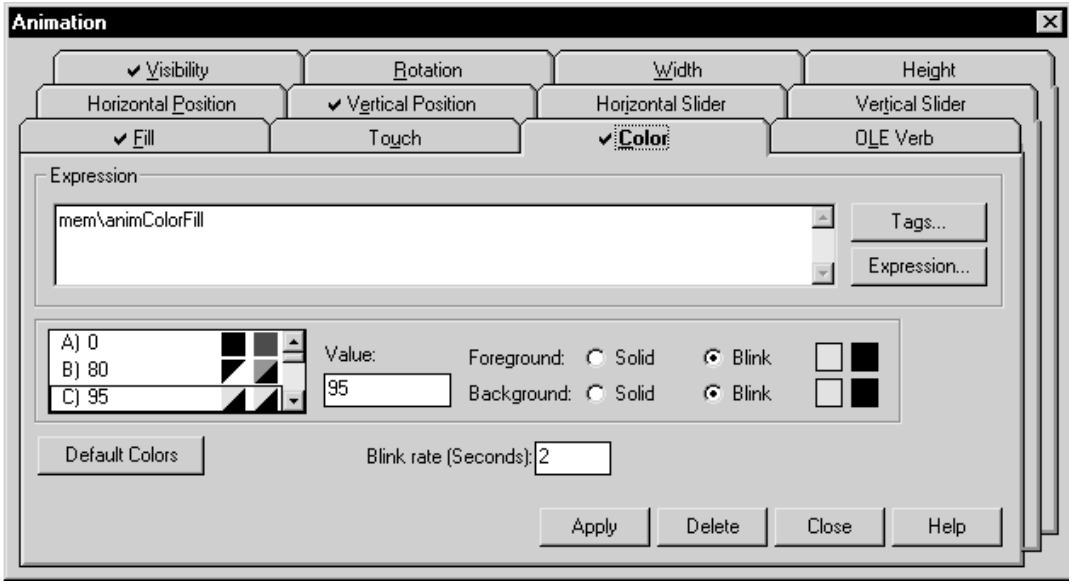
To view the animation on an object using the Animation menu

1. Select an object.
2. View the animation by doing one of the following:
   - Click the Animation menu and see which items have a check mark.
   - Right-click the object and then select Animation to see which items have a check mark.

To view the animation on an object using the Animation dialog box

1. Select an object.
2. On the View menu, click Animation.

When the Animation dialog box opens, look at which tabs have check marks on them to see which types of animation have been set up.



Copying or duplicating objects with animation

You can copy or duplicate objects that have animation attached to them. When you do, the animation attached to the objects is also copied or duplicated. If you copy or duplicate a group, the copy of the group can be ungrouped to individual objects, just like the original.

For information about copying and duplicating objects, see pages 20-42 and 20-43.

Copying animation without copying objects

If you have attached animation to an object, you can copy the animation and paste it onto another object. If the object has more than one type of animation, all animation is copied and pasted. Note that you can only copy animation to an object that supports the same type of animation.

To copy and paste animation

1. Select the object that has the animation you want to copy.
2. On the Edit menu, click Copy Animation, or right-click the object and then click Copy Animation.

3. Select the objects to copy the animation to.
4. On the Edit menu, click Paste Animation.

   To paste to a single object, you can right-click the object and then click Paste Animation.

Setting up animation for global objects

FactoryTalk® View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object. For information about creating base objects, see page 25-6.

The copies of the object are called reference objects, and they have the same properties as the original base object. If desired, you can assign unique animation to the reference object. For information about creating reference objects, see page 25-8.

The LinkAnimation property determines whether the reference object uses the base object's animation.

To set up animation for a reference object

1. Double-click the reference object to open the Property Panel.
2. Select the LinkAnimation property setting to use:
   - **Do not link**—allows you to set up separate animation for the reference object.
   - **Link with expressions**—the reference object uses the animation and tags or expressions assigned to the base object.
   - **Link without expressions**—allows you to use only the types of animation set up for the base object, but assign different tags or expressions to the reference object.
3. If you selected Do not link or Link without expressions, set up animation for the object using the methods described in this chapter.

For more information about setting up link properties and working with reference objects, see page 25-9.

23 Using expressions

This chapter describes:

- the types of expression components.
- using the Expression editor.
- which editors use expressions.
- formatting expressions.
- using tag names and tag placeholders with expressions.
- using constants.
- using operators.
- using math, security, and language functions.
- using if-then-else logic in expressions.
- the evaluation order of operators.
- using write expressions.

About expressions

Sometimes the data you gather from devices is meaningful only when you:

- compare it to other values.
- combine it with other values.
- create a cause-effect relationship with other values.

Expressions allow you to create mathematical or logical combinations of data that return more meaningful values. Depending on the components used in the expression, the value returned can be in the form of a numeric value, a true/false value, or a text string.

Expressions that result in floating-point values

If an expression results in a floating-point value but an integer value is required, the floating-point value is rounded.

For information about how values are rounded, see page 7-2.

Expression components

Expressions can be built from:

- tag values.
- tag placeholders.
- constants.
- arithmetic, relational, logical, and bitwise operators.
- mathematical and security functions.
- if-then-else logic.

Tags, arithmetic operators, bitwise operators, and mathematical functions such as SQRT (square root) return numeric values.

Relational and logical operators return true/false values. The security function CurrentUserHasCode(*x*) also returns a true/false value.

The security function CurrentUserName( ) returns a string value. The language function CurrentLanguage( ) returns a string value.

Expressions that use if-then-else logic can return numeric values, true/false values, or text strings, depending on how they are structured. These are called conditional expressions because the result of the expression depends on whether the If statement is true or false. When the If statement evaluates to true, the result is defined by the Then statement. When the If statement is false, the result is defined by the Else statement.

The Objects 5 Screen Demo sample application contains many examples of expressions. For example, see the alarm trigger expressions in the Alarm Setup editor.

Using the Expression editor

To create an expression, you can:

- type it directly in the "Tag or expression" column, for any connection that accepts expressions, or in the Expression box (for animation).
- open the Expression editor, and then create the expression in the editor.

Using the Expression editor versus typing expressions directly

Once you are familiar with expression syntax, you might find it quicker to create short expressions by typing them directly in the "Tag or expression" column.

The Expression editor allows you to see more text at once, which is useful for longer, more complicated expressions. Also, you can click buttons to enter tag names, operators,

and functions, thus avoiding typing mistakes. Another advantage of using the Expression editor is that you can check whether the syntax of the expression you've created is valid.

To create an expression by typing it directly

1. Type an expression up to 16,000 characters long.

Expressions that you type directly are not checked for syntax.

To open the Expression editor, do one of the following



Browse button in the Exprn column

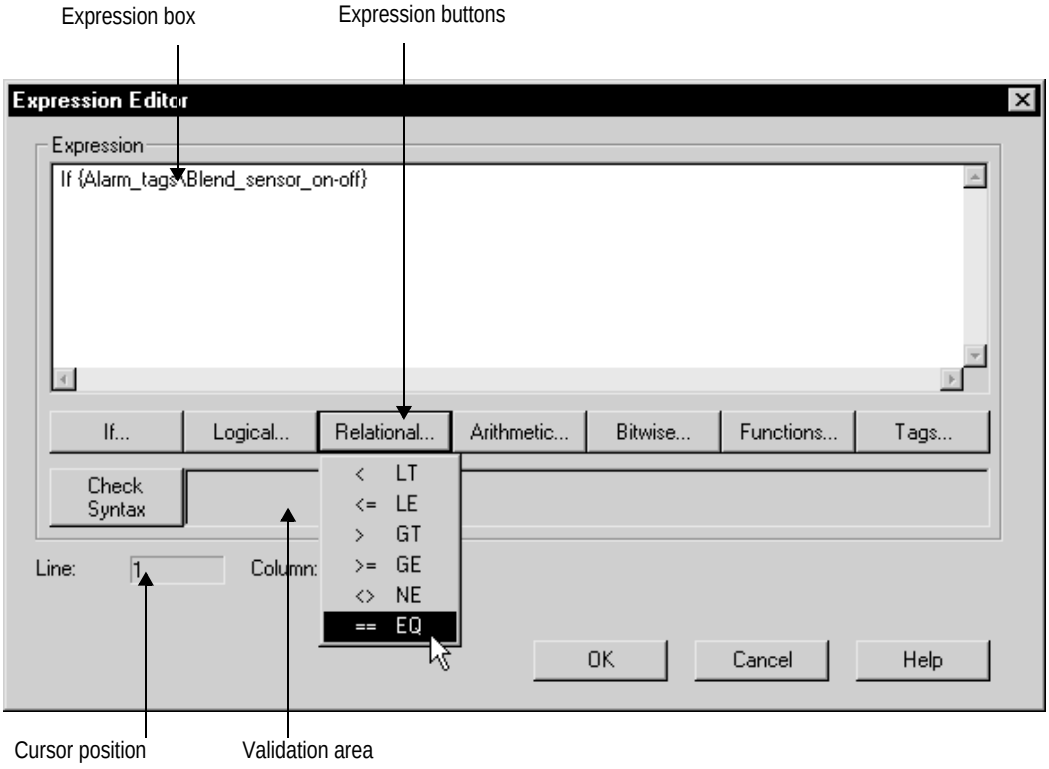
- Click the Browse button in the Exprn column for a connection that accepts expressions.

  The Browse button is not available for connections to which you can assign only tags.

- In the Animation dialog box, click the Expression button.

About the Expression editor

The Expression editor has these parts:



For details about using the options in the Expression editor, see Help.

Where you can use expressions

You can use expressions in these editors:

- Graphics—You can define an expression to control various aspects of a graphic object's appearance. For more information about assigning expressions to graphic objects, see page 20-38.

  You can also use expressions to attach animation to graphic objects. For more information, see Chapter 22.

- Global Connections—You can use expressions to remotely control when to open and print displays, as well as the date and time to display. For more information, see Chapter 8.
- Alarm Setup—When setting up alarms, you can use expressions for alarm triggers, and with some of the connections that silence and acknowledge alarms. For more information, see Chapter 9.
- Information Setup—You can use expressions to determine when to display information messages. For more information, see Chapter 27.
- Macros—You can use expressions in macros to assign values to tags. For more information, see page 30-1.

Formatting expressions

You can format expressions so they are easier to read. However, do not let tag names, function names, or function arguments span more than one line.

When formatting expressions, you can use line returns and multiple spaces.

Enclose strings in quotes. The string can contain any character, and can include spaces.

Example: Formatting an expression

To format this if-then-else statement, you can align the Else with the appropriate If, so the logic is easy to understand:

```
if (tag1 > tag2) then 0
else if (tag1 > tag3) then 2
else 4
```

Or you can condense it to the following:

```
if (tag1 > tag2) then 0 else if (tag1 > tag3) then 2 else 4
```

Using tag names and tag placeholders

A tag name can be included as part of an expression or can stand alone as the entire expression.

To supply a tag name, do one of the following:

- Type a tag name.

You can type a tag name that does not exist in the tag database. When you click OK, you are prompted to create the tag. You can create it now, or write down the name and create it later.

- Click the Tags button and select a tag from the Tag Browser.

Enclose tag names that contain dashes or start with a number in braces { } when you use them in an expression. This distinguishes the characters in the tag name from the characters in the expression.

You can use string tags as operands with the plus (+) arithmetic operator and with the relational operators.

Using tag placeholders instead of tag names

The Graphics editor accepts tag placeholders instead of tag names. Placeholders allow you to use the same display with different sets of tags.

You can use tag placeholders in:

- the graphic display that opens when the application is first run.
- graphic displays that are opened using a goto display button.
- graphic displays that are opened using a display list selector.

Use parameter files to specify which tags or folders to substitute for which placeholders. For global objects, you can specify the tags or folders of tags using global object parameters. For more information about using parameter files and global object parameters, see Chapter 25.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

To create a tag placeholder in an expression

1. Type the cross-hatch character followed by a number (no space in between). For example, #1.

Constants

A constant can have any of the following formats:

- integer (123)
- floating-point (123.45)
- string constant ("character string")

Arithmetic operators

Arithmetic operators perform math on two or more numeric values and calculate the result. The arithmetic operators are:

Symbol	Operator	Example (For these examples, tag1 = 5 and tag2 = 7)
+	addition	tag1 + tag2 returns a value of 12 You can also use this operator with string operands. See page 23-7.
-	subtraction	tag1 - tag2 returns a value of -2
*	multiplication	tag1 * tag2 returns a value of 35
/	division	tag1 / tag2 returns a value of 0.7142857
MOD,%	modulus (remainder)	tag2 MOD tag1 returns a value of 2 The modulus operator is the remainder of one number divided by another. In the example, the remainder of 7 divided by 5 is 2; so 7 % 5 = 2 **Important**: This operator is for integers only, not floating-point numbers.
**	exponent	tag1 ** tag2 returns a value of 78125

Be sure that any tag value you use as a divisor cannot at some point have a value of zero. Expressions that attempt to divide a number by zero produce an error at run time.

String operands

The + operator can be used to join string operands. For example, the expression “hello” + “world” returns: helloworld.

You cannot join string tags to analog tags, whether they are HMI or data server tags.

Relational operators

Relational operators compare two numeric or string values to provide a true or false result. If the statement is true, a value of 1 is returned. If false, 0 is returned.

The relational operators are:

Symbols	Operator	Numeric Example	String Example
For the numeric examples, tag1 = 5 and tag2 = 7 For the string examples, serial_no = "ST009"			
EQ, ==	equal	tag1 == tag2 false	serial_no == "ST009" true
NE, <>	not equal	tag1 <> tag2 true	serial_no <> "ST011" true
LT, <	less than	tag1 < tag2 true	serial_no < "ST011" true
GT, >	greater than	tag1 > tag2 false	serial_no > "ST011" false
LE, <=	less than or equal to	tag1 <= tag2 true	serial_no <= "ST011" true
GE >=	greater than or equal to	tag1 >= tag2 false	serial_no >= "ST011" false

How string operands are evaluated

String operands are evaluated by case and by alphabetical order. Lower case letters are greater than upper case letters. For example, h is greater than H. Letters later in the alphabet are greater than those earlier in the alphabet. For example, B is greater than A.

Logical operators

Logical operators determine the validity of one or more statements. There are three logical operators: AND, OR, and NOT. The operators return a non-zero value if the expression is true, or a 0 if the expression is false.

Any statement that evaluates to a non-zero value is regarded as true. For example, the statement tag1 is false if the value of tag1 is 0, and true if tag1 has any other value.

The logical operators are:

Symbols	Operator	Action	Example (For these examples, tag1 = 5 and tag2 = 7)
AND, &&	and	Returns a 1 if the statements to the right and left of the operator are both true.	(tag1 < tag2) AND (tag1 == 5) Both statements are true; returns a 1.
OR, ||	or	Returns a 1 if either the statement to the left or right of the operator is true.	(tag1 > tag2) OR (tag1 == 5) tag1 == 5 is true; returns a 1.
NOT	negation	Reverses the logical value of the statement it operates on.	NOT (tag1 < tag2) Although tag1 < tag2 is true, NOT reverses the logical value; returns a 0.

The parentheses are essential in the above expressions. They determine the evaluation order of the operators. For more information, see page 23-11.

Bitwise operators

Bitwise operators examine and manipulate individual bits within a value.

These operators are for integers only, not floating-point numbers. Do not use them with tags or expressions that return floating-point values.

Symbol	Operator	Action (for examples, see page 23-11)
&	And	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if both the corresponding bits in the original numbers are 1. Otherwise, the resulting bit is 0.
|	inclusive OR	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if either or both of the corresponding bits in the original numbers are 1. If both bits are 0, the resulting bit is 0.

Symbol	Operator	Action (for examples, see page 23-11)
^	exclusive OR (XOR)	Compares two integers or tags on a bit-by-bit basis. Returns an integer with a bit set to 1 if the corresponding bits in the original numbers differ. If both bits are 1 or both are 0, the resulting bit is 0.
>>	right shift	Shifts the bits within an integer or tag to the right. Shifts the bits within the left operand by the amount specified in the right operand. The bit on the right disappears. Either a 0 or a 1 is shifted in on the left, depending on whether the left-most bit is a 0 or a 1, and whether the operand consists of a signed or unsigned data type. For signed data types, if the left-most bit is 0, a 0 is shifted in. If the left-most bit is 1, a 1 is shifted in. In other words, the sign of the number is preserved. For unsigned data types, a 0 is always shifted in.
<<	left shift	Shifts the bits within an integer or tag to the left. Shifts the bits within the left operand by the amount specified in the right operand. The bit on the left disappears and 0 always shifts in on the right. See "Using the left shift operator," later in this chapter.
~	complement	Returns one's complement; that is, it toggles the bits within an integer or tag. Reverses every bit within the number so every 1 bit becomes a 0 and vice versa.

Using the left shift operator

If the left bit is a 1 an overflow occurs, and an error message is generated. To prevent this, use the bitwise AND operator with the left shift operator in an expression. For example:

(dev << 1) & 65535

where dev is a tag whose value is being shifted left, and 65535 is 1111 1111 1111 1111 in binary form.

Examples: Bitwise operators

For these examples, tag1 = 5 (binary 0000 0000 0000 0101) and tag2 = 2 (binary 0000 0000 0000 0010)

tag1 & tag2
Returns 0 (binary 0000 0000 0000 0000).

tag1 | tag2
Returns 7 (binary 0000 0000 0000 0111).

tag1 ^ tag2
Returns 7 (binary 0000 0000 0000 0111).

tag1 >> 1
Returns 2 (binary 0000 0000 0000 0010).

tag1 << 1
Returns 10 (binary 0000 0000 0000 1010).

~ tag1
Returns -6 (binary 1111 1111 1111 1010).

Evaluation order of operators

Expressions with more than one operator are evaluated in this order:

- Operators in parentheses are evaluated first.

  Therefore, to change the order of precedence, use parentheses.
- The operator with the highest precedence is evaluated next.
- When two operators have equal precedence, they are evaluated from left to right.

Operators are evaluated in this order:

Evaluation order	Symbols
1 (highest)	( )
2	NOT ~
3	* / MOD, % ** AND, && & >> <<
4	+ - OR, || | ^
5 (lowest)	EQ, == NE, <> LT, < GT, > LE, <= GE, >=

Examples: Evaluation order

For these examples, tag1 = 5, tag2 = 7, and tag3 = 10.

(tag1 > tag2) AND (tag1 < tag3)

is evaluated in this sequence:

1. tag1 > tag2 = 0
2. tag1 < tag3 = 1
3. 0 AND 1 = 0

The expression evaluates to 0 (false).

---

tag1 > tag2 AND tag3

is evaluated in this sequence:

1. tag2 AND tag3 = 1
2. tag1 > 1 = 1

The expression evaluates to 1 (true).

NOT tag1 AND tag2 > tag3 ** 2

is evaluated in this sequence:

1. NOT tag1 = 0
2. 0 AND tag2 = 0
3. tag3 ** 2 = 100
4. 0 > 100 = 0

The expression evaluates to 0 (false).

Mathematical functions

Use math functions to calculate the square root, log (natural or base 10), or trigonometry ratios (in radians or degrees) of a tag.

This function	Returns this value
SQRT (expression)	The square root of the expression
LOG (expression)	The natural log of the expression
LOG10 (expression)	The base ten log of the expression
SIN (expression)	The sine of the expression in radians
COS (expression)	The cosine of the expression in radians
TAN (expression)	The tangent of the expression in radians
ARCSIN (expression)	The arc sine of the expression in radians
ARCCOS (expression)	The arc cosine of the expression in radians
ARCTAN (expression	The arc tangent of the expression in radians
SIND (expression)	The sine of the expression in degrees
COSD (expression)	The cosine of the expression in degrees
TAND (expression)	The tangent of the expression in degrees
ARCSIND (expression)	The arc sine of the expression in degrees
ARCCOSD (expression)	The arc cosine of the expression in degrees
ARCTAND (expression)	The arc tangent of the expression in degrees

Security functions

Use security functions to control access to your application.

These functions allow you to determine a user's identity or security rights in order to limit access to the application based on these criteria.

This function	Returns this value
CurrentUserHasCode (*Security Code Letters*)	True (1) if any of the specified security codes have been assigned to the user; false (0) if not. If checking multiple security codes, do not type a space between the security code letters. For example: CurrentUserHasCode (ABP) returns the value 1 if the user has been assigned one or more of the specified codes.
CurrentUserName( )	A string containing the name of the current user. This function is case sensitive. All RSView® 3.20 and earlier user names use uppercase letters.

For more information about setting up security for your application, see Chapter 11.

For an example of using the CurrentUserHasCode(*x*) function, see page 11-11. For examples of using the CurrentUserName( ) function, see page 11-24.

Language function

The language function shows you which language your application is currently using.

You can display the current language in a string display, or use it in expressions to generate language-specific messages for your users.

This function	Returns this value
CurrentLanguage( )	RFC1766 name of the current runtime language.

The RFC1766 name is a standard way of representing a language using the format: *languagecode-Country/RegionCode*

where *languagecode* is a lowercase two-letter code and *Country*/*RegionCode* is an uppercase two-letter code.

For example, U.S. English is en-US.

For more information about setting up languages for your application, see Chapter 12. For a list of RFC1766 names, see Appendix F.

Language switching alarm, information, and local messages in RSView ME Station 4.00

FactoryTalk View 5.00 provides direct support for switching message languages at run time. For applications that will run in RSView ME Station version 4.00, you can use the CurrentLanguage( ) expression function to set up language switching for messages. To do this, set up trigger value offsets for series of messages in the Alarm Setup editor, information message files, and local message files.

Export the text strings in the Alarm Setup editor and message files for translation. Then paste the translated strings into the editor and original message files, and assign each string the correct trigger value. For information about exporting text for translation, see Chapter 12.

Example: Setting up information messages in multiple languages

This example shows how to generate English, French, or German information messages at run time in an RSView ME Station 4.00 application, depending on which language the application is using.

1. Create a tag called *Information_messages* that will generate trigger values of 11 to 20 for different conditions that require information messages.
2. Create an information message file.
3. Create English messages for trigger values 11 to 20.

   Tip: Messages are sorted alphanumerically in the Excel spreadsheet or Unicode text file created for translation. Therefore, numbers 2 through 9 would appear after 10, 11, 12, and so on. To keep your messages in order in the translation file, begin the first series of numbers at 11.
4. Create French messages for trigger values 21 to 30.
5. Create German messages for trigger values 31 to 40.
6. Assign this expression to the Value connection in the Information Setup editor:

   ```
   If CurrentLanguage( )="en-US" then Information_messages
   Else If CurrentLanguage( )="fr-FR" then Information_messages + 10
   Else Information_messages + 20
   ```

If-then-else

If-then-else expressions carry out an action conditionally or branch actions depending on the statements in the expression. The if-then-else statements enable the expression to perform different actions in different situations and to repeat activities until a condition changes.

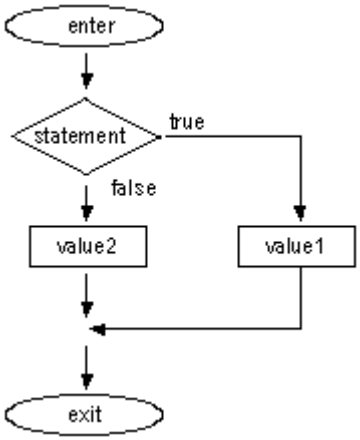
To build conditional expressions, use the relational operators and the logical operators for the statement and values.

The if-then-else structure is:

if *statement* then *value1* else *value2*

If the statement is true then the expression returns value1; if the statement is false then the expression returns value2. If the result of the statement is a non-zero value, the statement is true (and returns value1); if the result is 0, the statement is false (and returns value2).

The if-then-else structure is illustrated here.



Nested if-then-else

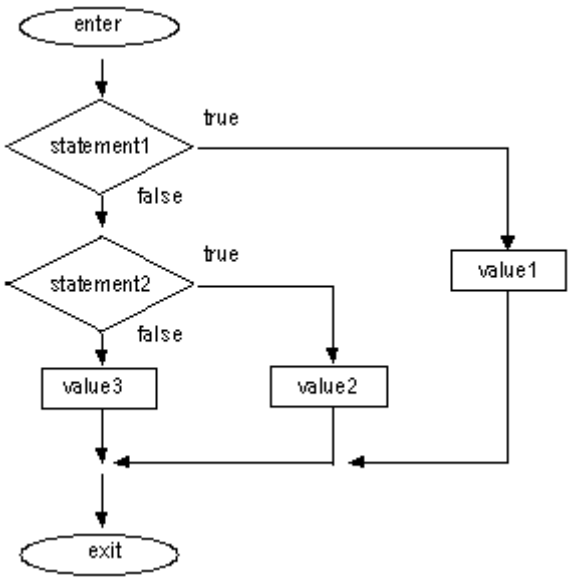
You can also nest an if-then-else structure inside the Then or Else part of an if-then-else structure.

Example 1: Nested if-then-else

This expression:

if *statement1* then *value1*
else if *statement2* then *value2*
else *value3*

has this interpretation:

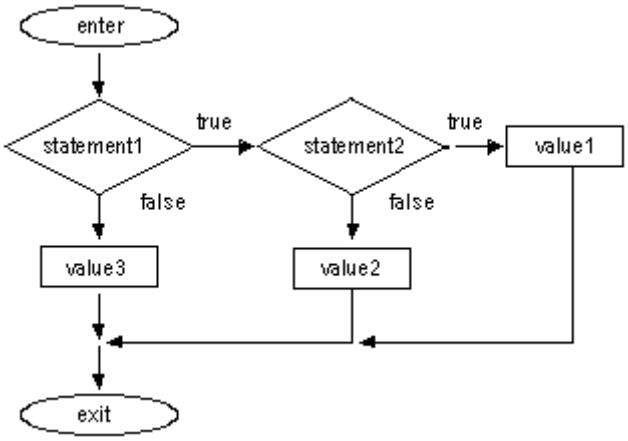


Example 2: Nested if-then-else

This expression:

if *statement1* then
if *statement2* then value1
else *value2*
else *value3*

has this interpretation:



Using write expressions

Write expressions allow the operator to enter a value which is manipulated by an expression before being sent to the data source. FactoryTalk® View substitutes the value the operator enters for the placeholder in the expression, calculates the value of the expression, and writes the result to the Value connection. All write expressions must contain a question mark (?) as a placeholder for the value the operator enters.

You can use write expressions with the numeric input enable button and the numeric input cursor point. When the operator presses the button or cursor point, a keypad or scratchpad opens. The operator enters a value in the keypad or scratchpad, and this value is substituted for the ? placeholder in the write expression.

Example: Using write expressions

In this example, the operator regulates the speed of a conveyor belt by entering a value in feet or meters per second. When the operator enters the value in meters per second, the value is converted to feet per second before being passed to the data source.

The operator first indicates whether the value is in feet or meters by pushing a maintained push button. The push button has one state corresponding to feet per second, and the other state to meters per second.

Then the operator presses the numeric input enable button and enters the value for the conveyor speed in a numeric pop-up keypad. The ? character in the write expression is the placeholder for the value the operator enters.

To set up the maintained push button

1. In the Maintained Push Button Properties dialog box, in the States tab, set up these states:
   - State 0—Value: 0, Caption: Feet/S
   - State 1—Value: 1, Caption: Meters/S
2. In the Connections tab, assign a digital tag called Feet_or_meters to the Value connection (either an HMI tag or a data server tag).

To set up the numeric input enable button

1. In the Numeric Input Enable Properties dialog box, in the Label tab, type the caption "Enter conveyor speed".
2. In the Connections tab, assign a tag called Conveyor_speed to the Value connection.
3. Assign this expression to the Optional Exp connection:

   if Feet_or_meters == 0 then

   ?

   else

   ? * 3.281

FactoryTalk View writes the result of the expression to the Conveyor_speed tag at the data source.

24 Using embedded variables

This chapter describes:

- the types of embedded variables.
- where you can create embedded variables.
- creating embedded variables.
- embedded variable syntax.
- how embedded variables are updated at run time.
- how embedded variables are displayed at run time.

About embedded variables

Embedded variables allow you to display values that change dynamically at run time. You can use embedded variables in the text captions on graphic objects, in the title bar of On Top displays, and in message text. You can use multiple embedded variables in the same caption or message.

For example, you could embed a tag value and the time variable in a local message. At run time when the local message is displayed, it is updated to reflect the tag's current value as the value changes. The time is also updated as the time changes.

You can also use literal strings and constants in embedded variables, or a combination of both variable and literal strings and numbers.

Embedded variables can consist of:

- numeric (analog or digital) tags, including both HMI and data server tags.
- literal numbers (constants).
- string tags, including both HMI and data server tags.
- literal strings (static text).
- tag placeholders. For information about tag placeholders, see page 20-39.
- the time.
- the date.

Where you can create embedded variables

You can create embedded variables in these editors:

- Graphics—Use this editor to insert embedded variables in the captions for graphic objects. For graphic objects with multiple states, you can insert different embedded variables in each state's caption.

  You can also use embedded variables in the title bar for On Top displays.

  For information about specific graphic objects, see Chapter 21. For information about On Top displays, see page 19-13.

- Local Messages—Use this editor to insert embedded variables in local messages.

  For more information about local messages, see page 19-25.

- Information Messages—Use this editor to insert embedded variables in information messages.

  For more information about information messages, see Chapter 27.

- Alarm Setup—Use this editor to insert embedded variables in alarm messages.

  For more information about alarms, see Chapter 9.

Creating embedded variables

To create an embedded variable in a graphic object's caption

1. Open the graphic object's Properties dialog box.
2. Click the tab containing the Caption box.

   The Caption box is on the Label tab or the States tab, depending on the type of object.

   For text objects, use the Text box on the General tab.
3. Click Insert Variable.
4. Click the type of variable to insert.
5. Fill in the options in the dialog box that opens. For details about the options, see Help.

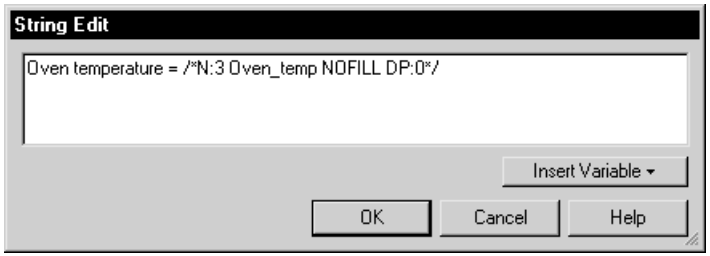
To create an embedded variable in an On Top display's title bar

1. Open the Display Settings dialog box.
2. Select the Title Bar check box if it is not already selected.
3. Click Insert Variable.
4. Click the type of variable to insert.

5. Fill in the options in the dialog box that opens. For details about the options, see Help.

To create an embedded variable in a message

1. In the Message column of the Local Messages, Information Messages, or Alarm Setup editor, right click and then click Edit String.



2. Click Insert Variable.
3. Click the type of variable to insert.
4. Fill in the options in the dialog box that opens. For details about the options, see Help.

Embedded variable syntax

Embedded variables are case sensitive, and must use specific syntax to work. Otherwise, the embedded variable is treated as a piece of text. Therefore, we do not recommend creating and editing embedded variables manually. Instead, use the Insert Variable and Edit Variable dialog boxes.

Numeric embedded variable syntax

Use numeric embedded variables to insert analog or digital tag values into captions, title bars, and messages. You can use both HMI and data server tags.

You can also insert “literal” numbers to display a constant or to specify a tag placeholder in the caption or message.

Numeric embedded variables use this syntax:

/*LN:# *Tag_name Fill_character* DP:#*/

where

L (optional) indicates it is a literal (constant) number. This symbol prevents a tag read. If you type a tag placeholder for the *Tag_name*, the value of the placeholder is substituted from the parameter file or global object parameter definition.

N indicates it’s a numeric embedded variable.

indicates the number of digits.

Tag_name is the tag to display; you can also type a literal number or tag placeholder here.

Fill_character is the fill character to use: NOFILL, ZEROFILL, or SPACEFILL.

indicates the number of decimal places.

Examples: Numeric embedded variable syntax

To display the current value of a tag called Oven_temp, with 3 digits, no decimal places, and no fill character, type this:

/*N:3 Oven_temp NOFILL DP:0*/

To display the constant 48, with 3 decimal places and 2 zeroes to the left of the number (for a total length of 8 digits including the decimal), type this:

/*LN:8 48 ZEROFILL DP:3*/

At runtime the numeric embedded variable would look like this: 0048.000.

String embedded variable syntax

Use string embedded variables to insert string tag values into captions, title bars, and messages.

You can also insert "literal" strings of static text. For example, you can type a word or phrase, a tag placeholder, or a number. To control how constant numbers are displayed, use a literal numeric variable rather than a string variable.

String embedded variables use this syntax:

/*LS:-# *Tag_name* SHOWSTAR*/

where

L (optional) indicates it is a literal (static) string. This symbol prevents a tag read. If you type a tag placeholder for the *Tag_name*, the value of the placeholder is substituted from the parameter file or global object parameter definition. The value cannot contain spaces.

S indicates it's a string embedded variable.

- (optional). A minus sign (-) before the # indicates that if the string is longer than the fixed number of characters, the right-most characters will be displayed.

indicates the number of characters if you select a fixed number of characters; type 0 if you don't want to use a fixed number.

Tag_name is the tag to display; you can also type a literal string or tag placeholder here. The string cannot contain spaces.

SHOWSTAR (optional) specifies that if the string is longer than the fixed number of characters, an asterisk (*) will be displayed to indicate the string is truncated. If you use the minus sign (-), the asterisk will be displayed as the first character at the left end of the string. Otherwise, the asterisk will be displayed as the last character at the right end of the string.

Examples: String embedded variable syntax

To display the current value of a string tag called Blower_status, with a fixed length of 20 characters, you would type this:

/*S:20 Blower_status*/

To display the literal string Oven temperature, type this:

/*LS:20 “Oven temperature”*/

To display the literal string 36.5, type this:

/*LS:3 36.5*/

To assign the tag placeholder #1, without a fixed string length, type this:

/*S:0 #1*/

To display the value of the string tag Conveyor_message, with a fixed length of 40, displaying the right-most characters, with an asterisk to indicate if the message is truncated, type this:

/*S:-40 Conveyor_message SHOWSTAR*/

Time and date embedded variable syntax

Use time and date embedded variables to insert the current time or date into captions, title bars, and messages

Time and date embedded variables use this syntax:

/*_Time_date_format_*/

where

Time_date_format uses one of these character sequences:

These characters	Specify this format
SD	Short date
LD	Long date
SDT	Short date and time
LDT	Long date and time
T	Time
TSD	Time and short date
TLD	Time and long date

Example: Time and date embedded variable syntax

To display the time followed by the short date, you would type this:

/*TSD*/

A space is placed between the time and date when the embedded variable is displayed at run time.

How embedded variables are updated at run time

At run time, this is how embedded variables are displayed and updated:

- Graphic objects and title bars—When a display containing a graphic object or title bar that uses an embedded variable is open, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.

- Local messages—When a display containing a local message display object is open, and the message the object is displaying contains an embedded variable, the value of the embedded variable is updated whenever a new tag value is read from the data source. For time and date embedded variables, the time and date are updated as the system time and date change.
- Information messages—The value of the embedded variable is read when the information message is first displayed. It is not updated after that.

  If the message is printed, it is printed using the value the variable had when the message was first displayed. This value is retained if you shut down and restart the application.
- Alarm messages—The value of the embedded variable is read when the alarm occurs, and is displayed in the message associated with the alarm. It is not updated after that.

  If the message is printed, it is printed using the value the variable had when the alarm first occurred. This value is retained if you shut down and restart the application.

How embedded variables are displayed at run time

If there is no valid data available for the embedded variable, the variable is replaced with question marks (?). This could occur when a display first opens and the data has not arrived yet, or when there is a problem that prevents communication with the data source.

If a string or numeric embedded variable has been set up but no tag has been assigned, the embedded variable is replaced with asterisks (*).

Numeric embedded variables

The value shown for a numeric embedded variable depends on whether the tag value is a floating-point number or an integer. Integer values are displayed as is. Floating-point values are rounded to fit the specified number of digits for the variable.

For example, if the variable is set up to show 6 digits, 1234.56 is rounded to 1234.6. 1234.44 is rounded to 1234.4. The decimal counts as one of the digits.

For more information about how values are rounded, see page 7-2.

If the tag value, including the decimal point and minus sign, contains more digits than specified for the variable, the numeric variable is replaced with asterisks.

For example, if the variable is set up to show 6 digits, and the tag value is -123456, the variable will be replaced with asterisks.

Literal numbers are displayed using the same rules as numeric tag values.

Number formats

The numeric variable uses the number format of the current application language. For example, if the application language uses a comma for the decimal symbol, the numeric variable uses a comma for the decimal symbol.

For information about using multiple languages, see Chapter 12.

String embedded variables

For string embedded variables that do not use a fixed number of characters, the entire string tag value is displayed, unless a null character is read. Nothing after a null character is displayed.

If a fixed number of characters is used, the variable displays the value of the tag up to the number of characters specified, unless a null character is encountered before the specified length. Nothing is displayed after a null character. If necessary, spaces are used to make up the required number of characters.

Null characters have a hex value of 0. The null character indicates the end of string input. It does not add to the actual string length.

If the string is longer than the specified number of characters, it is truncated to fit the number of characters. If the SHOWSTAR option is used, an asterisk (*) replaces the first or last character displayed. If the embedded variable is set up to display right-most characters, excess characters are truncated at the left end of the string and the asterisk (if used) appears at the left. Otherwise, the right-most characters are truncated and the asterisk (if used) appears at the right.

Literal strings are displayed using the same rules as string tag values.

Time and date embedded variables

For embedded variables that show both the time and the date, a space is placed between the time and date when the embedded variable is displayed at run time.

Time and date formats

Time and date embedded variables use the time and date formats for the current application language. For example, if you specify the short date format, at run time the display uses the short date format that the application language uses.

For information about using multiple languages, see Chapter 12.

25 Using parameters and global objects

This chapter describes:

- using tag placeholders and parameter files.
- creating tag placeholders.
- creating parameter files.
- using global objects.
- creating global object displays and base objects.
- creating, setting up, and deleting reference objects.
- using global object parameters.
- adding process faceplates that connect to Logix5000 processors.

The topics in this chapter describe features of FactoryTalk® View Studio that can help you set up your applications more quickly by reusing similar groups of objects and graphic displays:

- Tag placeholders and parameter files allow you to use the same graphic display with different sets of tags.
- Global objects allow you to use multiple instances of the same graphic object and make changes to all of the objects at once.
- Global object parameters allow you to assign different sets of tags to different copies of the object without breaking the link to the base object, thus preserving the ability to update all copies of the object at once.
- Process faceplates provide pre-configured graphic displays that interact with the instructions in Logix5000 processors.

Using tag placeholders and parameter files

To use the same graphic display with different sets of tags, use tag placeholders and parameter files. Using tag placeholders can be quicker than duplicating a display and setting up the objects in it to use a different set of tags, especially when the display uses a lot of tags. Using parameter files also reduces the size of the runtime application file.

Tag placeholders can provide a way to use one graphic display to represent a number of similar operations. For example, suppose you are creating displays for a plant that cans corn and peas. The machinery used in both processes is identical. Instead of creating two displays and specifying corn-related tags in one display and pea-related tags in another,

you can create one display and not specify any tag names. Where a tag name is required, type a tag placeholder.

Use parameter files to specify which tags to substitute for which placeholders. For information about using parameter files, see page 25-3.

You can use tag placeholders wherever you would normally assign a tag to an object, including in expressions and embedded variables. You can also use tag placeholders in the expressions you create to animate objects.

You can also use tag placeholders with global objects. You can assign tag placeholders to the connections for base objects, and to connections for reference objects with the LinkConnections property set to False. For more information about global objects, see page 25-6.

You can use tag placeholders in:

- the graphic display that opens when the application is first run. Specify the graphic display to open, and the parameter file to use with it, in the Startup editor.
- graphic displays that are opened using goto display buttons. Specify the graphic display to open, and the parameter file to use with it, when you set up the button.
- graphic displays that are opened using display list selectors. Specify the graphic displays to open, and the parameter files to use with them, when you set up the display list selector.

Summary of steps

Follow these steps to use tag placeholders and parameter files:

1. In the Graphics editor, create graphic objects and assign tag placeholders to the objects.
2. In the Parameters editor, create parameter files for each set of tags that the display will use. In the parameter files, specify which tags to substitute for which placeholders.
3. In the Graphics editor, create goto display buttons or display list selectors for opening the display containing tag placeholders. Specify the appropriate parameter files in the Goto Display Button Properties dialog box or Display List Selector Properties dialog box.

   For information about setting up goto display buttons, see page 21-35. For information about setting up display list selectors, see page 21-37.
4. If the startup display uses tag placeholders, in the Startup editor specify the parameter file to use with the startup display.

   For information about the Startup editor, see Chapter 14.

Creating tag placeholders

A tag placeholder is the cross-hatch character (#) followed by a number.

The tag placeholder can replace any part of a tag name, including folder names. For example, you could create a parameter file specifying that the tag placeholder #1=Folder1. You could assign the folder and a tag name to a graphic object's connection: #1\Tag1.

You can create tag placeholders in:

- the Connections tab of an object's Properties dialog box.
- the Connections tab of the Property Panel.
- the Expression box in the Animation dialog box.
- anywhere that you can insert an embedded variable. For information about embedded variables, see Chapter 24.

To create a tag placeholder

1. Type the cross-hatch character followed by a number (no space in between). For example, #1.

Creating parameter files

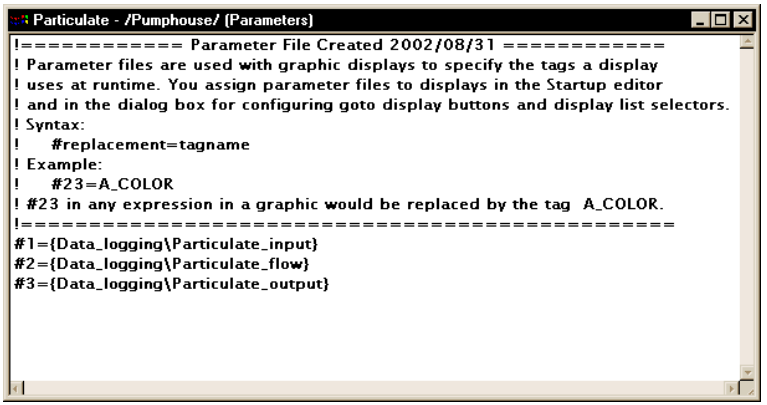
The parameter file specifies which tags to substitute for the placeholders in the display, by assigning one tag to each unique placeholder in the display. Create a parameter file for each set of tags that you want to use with the same graphic display.

At run time, the tag values that are displayed depend on which parameter file is used when the display opens. When you open the display, the tags specified in the parameter file replace the tag placeholders.

Parameters can replace any portion of a tag address. For example, you can use parameters to replace folder names.

Using the Parameters editor

Use the Parameters editor to create one or more files of tag placeholder replacements. Each file is stored in the editor's folder. You can open and work on multiple parameter files at the same time.


Particulate - /Pumphouse/ (Parameters)
!============ Parameter File Created 2002/08/31 ============
! Parameter files are used with graphic displays to specify the tags a display
! uses at runtime. You assign parameter files to displays in the Startup editor
! and in the dialog box for configuring goto display buttons and display list selectors.
! Syntax:
! #replacement=tagname
! Example:
! #23=A_COLOR
! #23 in any expression in a graphic would be replaced by the tag A_COLOR.
!==
#1={Data_logging\Particulate_input}
#2={Data_logging\Particulate_flow}
#3={Data_logging\Particulate_output}

For details about using the Parameters editor, see Help. To open the Help, press F1 on your keyboard.

Example: Using a parameter file to replace tag placeholders

This example shows how to use a graphic display called Canning Overview with two sets of tags, one for canning corn, and one for canning peas.

The Canning Overview display is opened from a graphic display called Main Menu.

1. Create these sets of tags. You can use both HMI and data server tags.

Tag type	Tag name	Tag name
String	Corn_Name	Pea_Name
Analog	Corn_Temp	Pea_Temp
Analog	Corn_Weight	Pea_Weight
Analog	Corn_Level	Pea_Level

2. Create two parameter files, called Corn and Peas, containing these parameters:

Corn	Peas
#1=Corn_Name	#1=Pea_Name
#2=Corn_Temp	#2=Pea_Temp
#3=Corn_Weight	#3=Pea_Weight

Corn	Peas
#4=Corn_Level	#4=Pea_Level

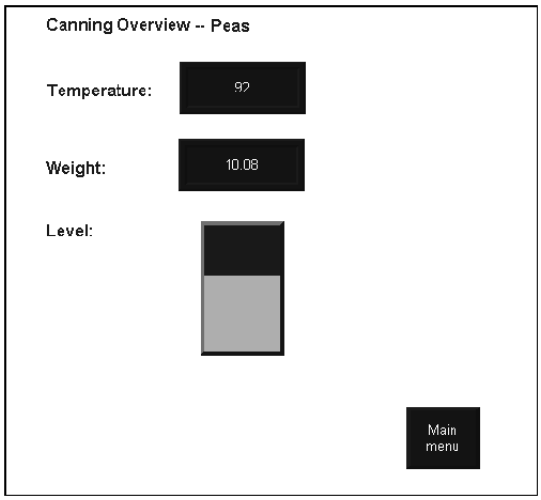
3. In the Canning Overview display, assign tag placeholders to the Value connections for these graphic objects:

This graphic object	Uses this tag placeholder
String display	#1
Numeric display 1	#2
Numeric display 2	#3
Bar graph	#4

4. Use descriptive text to illustrate the objects in the display.
5. In the Main Menu display, create a display list selector for opening the Canning display, with two states.
6. Assign the Corn parameter file to one state, and the Peas parameter file to the other.
7. Call the captions for the states Corn and Peas, respectively.

At run time, when the operator selects the Peas state on the display list selector in the Main Menu, and presses the Enter key, the Canning Overview display opens and shows the values of the Pea_ tags. When the operator selects the Corn state, the values of the Corn_ tags are displayed.

The Canning Overview display looks like this when the operator selects the Peas state:



Using global objects

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When you update the original object, the changes are automatically applied to all the copies of the object.

You create global objects in global object displays, in the Global Objects folder of the Explorer window. The objects you create in a global object display are called base objects.

You can copy or drag base objects into standard graphic displays. The copied object is called a reference object. You can copy a global object into any number of graphic displays, and multiple times into the same graphic display.

Global object displays have the file extension .ggfx. They are stored in this default location:

\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects*Application Name*\Global Objects (Windows® 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME\HMI projects*Application Name*\Global Objects (Windows XP or Windows Server 2003 R2)

where *Application Name* is the name of your application.

Summary of steps

Follow these steps to set up global objects:

1. In the Graphics editor, create a global object display. For more information, see the next section.
2. Create graphic objects in the global object display. These are the base objects. Set up their appearance, animation, and connections.
3. Copy or drag base objects into a standard graphic display. The copied objects are reference objects. For more information, see page 25-8.
4. Edit the link properties of the reference objects as desired. For more information, see page 25-9.

Creating global object displays and base objects

You can create a global object display in the Global Objects folder, or create a graphic display in the Displays folder and then add the display to the Global Objects folder. You can also add library displays to the Global Objects folder.

The objects you create in the global object displays are called base objects.

To create a global object display

1. In the Graphics folder, right-click Global Objects, and then click New.
2. Create graphic objects in the display. These are the base objects.

   You can not use ActiveX objects as base objects. You cannot convert objects to wallpaper in the global object display.

   For information about creating graphic objects, see Chapter 20.



Save tool

3. On the File menu click Save, or click the Save tool.
4. In the Component name box, type a name for the display, and then click OK.

   The display is added to the list in the Global Objects folder.

To add a display or library to the Global Objects folder

1. In the Explorer window, in the Graphics folder, right-click the Global Objects icon.
2. Click Add Component Into Application.
3. Browse to and select the display or library to add, and then click Open.

The display is copied into the Global Objects folder, and given the file extension .ggfx.

Any ActiveX objects are deleted. All other objects are converted to base objects.

If you delete, remove, or rename a global object display, you break the links between the base objects in the display and their reference objects. For more information about breaking links, see Help.

Creating reference objects

FactoryTalk View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object. The copies of the base object are called reference objects.

You can copy or drag base objects into standard graphic displays. Each copied object becomes a reference object. You can also copy, drag, and duplicate reference objects that you have already created to create more copies of the base object.

Base objects can be group objects. This provides powerful template capabilities. When you add or remove objects from the base object group, all the reference objects are automatically updated.

If desired, you can assign unique connections, animation, and size to the reference object.

To create a reference object

1. Copy an object from a global object display to a standard graphic display.
2. To assign unique connections, animation, or size to the object, double-click the object to open the Property Panel.

   For information about using the Property Panel, see page 20-29.

You can also create reference objects by adding a global object display to the Displays folder. All the objects in the new display are reference objects.

To add a global object display to the Displays folder

1. In the Explorer window, in the Graphics folder, right-click the Displays icon.
2. Click Add Component Into Application.
3. Browse to and select the global object display to add, and then click Open.

   The display is copied into the Displays folder, and given the file extension .gfx. All the objects are converted to reference objects.

To edit a reference object's base object

1. Right-click the reference object, and then click Edit Base Object.
2. The global object display containing the base object opens, with the object selected.
3. Make your changes to the base object.

   All reference objects linked to the base object are updated.

To view the changes to the reference object, close the display containing the reference object, and then reopen it. Or, toggle the object's LinkSize property on and off.

Setting up reference objects' link properties

FactoryTalk® View global objects allow you to link the appearance and behavior of a graphic object to multiple copies of that object. When the original base object is updated, the changes are automatically applied to all the copies of the object.

The copies of the object are called reference objects, and they have the same properties as the original base object. If desired, you can assign unique size, connections, and animation to the reference object.

To assign unique properties to the reference object, use the Property Panel to edit the following properties for the reference object:

- LinkSize—select False to set up the height and width separately for the reference object. If LinkSize is set to true, when you try to resize the reference object, it will snap back to its linked size.
- LinkConnections—select False to set up connections separately for the reference object. Whether you use the base object's connections, or set up separate connections, the tags used count towards the tag limit for the display.

  You can assign tag placeholders to the connections for base objects, and to connections for reference objects with the LinkConnections property set to False.
- LinkAnimation—select Do not link to set up separate animation for the reference object. To use the animation set up for the base object but assign different expressions to the reference object, select Link without expressions.

  If you select Link with expressions, the reference object uses the animation and expressions assigned to the base object.

Reference objects also have a property called LinkBaseObject, which specifies the name and location of the base object to which the reference object is linked. You cannot edit the LinkBaseObject property.

To change a reference object's link properties

1. Double-click the object to open the Property Panel.
2. Make your changes to the link properties.
3. Specify the new size, connections, or animation for the object.

   For information about resizing objects, see page 20-44. For information about assigning connections to objects, see page 20-31. For information about assigning animation to objects, see Chapter 22.

To specify default link properties for reference objects

1. On the Edit menu, select Global Object Defaults.
2. Specify the new default values.
3. Click OK.

   The defaults are used for any new reference objects you create. You can still edit the properties for individual objects separately.

To break the link to a base object

1. Right-click the reference object, and then click Break Link.

   The reference object becomes a regular object. You can edit all of its properties. However, you cannot restore the connection to the base object.

Deleting the base object

If you delete a base object or a global object display containing base objects, any reference objects that are linked to the deleted base objects are broken. A broken reference object is displayed as a red square with an X through it. You cannot edit broken reference objects.

To repair a broken reference object

1. Recreate the base object with the same object type and name as before, on the same display as before.

Using global object parameters

Global object parameters are parameters that you can assign to global objects. A global object parameter allows you to assign different tags or sets of tags to each reference object without breaking the link to the base object. This allows you to make changes to the base object and all the associated reference objects at the same time.

For example, the Logix_PIDE global object display contains a grouped object composed of other grouped objects. The objects' connections are set up with tags and expressions

that use values from a set of Logix5000 tags. When you create multiple reference objects from this base object, each reference object can use a different set of Logix5000 tags. The global object parameters you assign to the base object allow you to do this, because you are using a placeholder instead of a specific backing tag (a backing tag is a path to a folder of tags; it is also known as a structured tag). If you then change, add, or remove a tag or expression in the base object, the same change is made to all the reference objects.

If you don't use global object parameters, you can still assign different tags to different reference objects by changing the reference objects' LinkConnections property to False and the LinkAnimations property to Link without expressions.

The global object parameter takes the same form as a regular parameter: #*1*, where *1* can be any number from 1 to 500. The parameter can be the placeholder for an individual tag or for a folder of tags. For example, #1 could be a placeholder for the path to the folder containing the PIDE tags assigned to the global object.

When you set up the base object, specify the global object parameters to use with the object. You can provide a description of each parameter to remind you or another application designer of the type of value to assign to the parameter on the reference object. Then assign specific values to each parameter for the reference object. You can assign numeric or string constants, tags, or backing tags.

Difference between global object parameters and regular parameters

Global object parameters allow you to assign different values to different instances of the same placeholder. For example, each reference object in the display might have the placeholder #1. Using regular parameter files, you could only assign one value to #1, and this would apply to all objects in the graphic display. With global object parameters, you can assign a different value to #1 for each object that uses the placeholder #1.

If a placeholder is defined in a global object parameter for an individual object and defined in a parameter file, the definition assigned to the object takes precedence.

Using global object parameters with group objects

To use global object parameters with a group object, the parameters are defined at the group level, not at the level of individual objects. Thus the definition applies to each object within the group. You can assign as many parameters as desired to the group. For example, you might assign #1 to some members of the group and #2 to other members of the group. If you create a global object parameter definition for an object and later group the object, the definition is deleted. In addition, if you create a global object parameter definition for a group and then ungroup the object, the definition is deleted.

Values are assigned to the global object parameters at the group level as well. You cannot assign separate values to individual objects in the group.

Summary of steps

Follow these steps to use global object parameters:

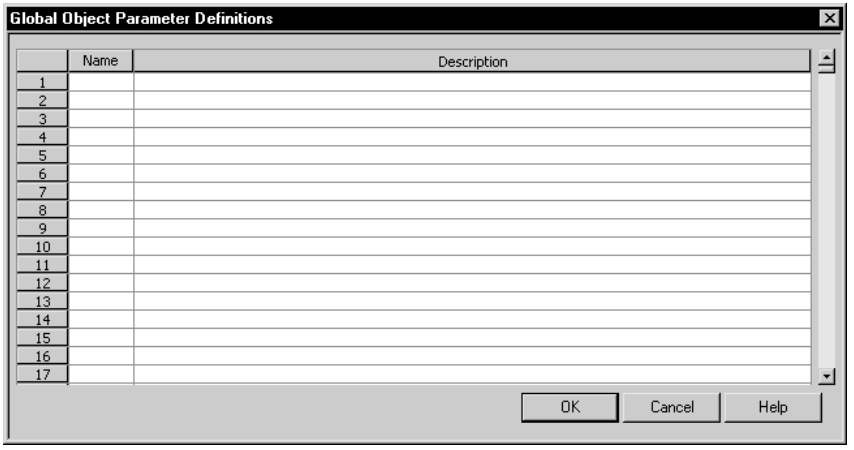
1. In the global object display, assign placeholders and parameter definitions to the base object. For details, see the next section.
2. In the standard graphic display, select the reference object and specify the value to use for each parameter. For more information, see page 25-12.

To specify global object parameters for the base object

1. Create placeholders in each place where you want to use a global object parameter. You can create the placeholders anywhere a tag or expression is required.

   You can also type placeholders in embedded variables by using a literal string embedded variable. For information about embedded variables, see Chapter 24.

2. Right-click the global object or grouped global object, and then click Global Object Parameter Definitions.

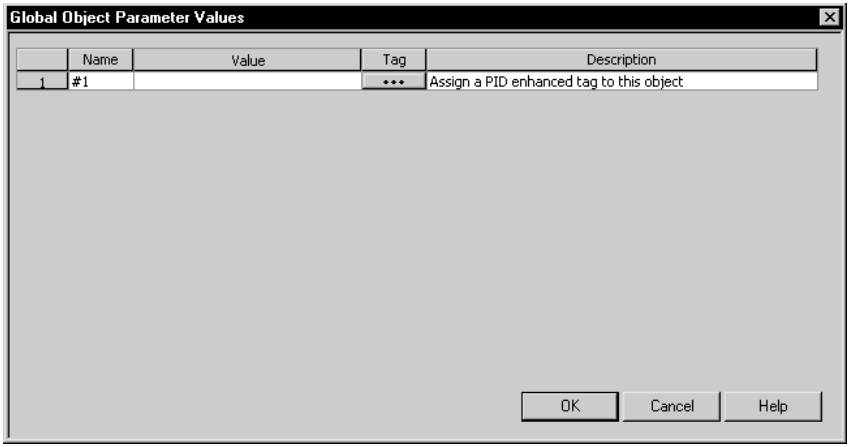


3. Specify a parameter for each placeholder for the object. If desired, type a description for each parameter.

   For details about using the Global Object Parameter Definitions dialog box, see Help.

To specify the parameter values for the reference object

1. Right-click the reference object or grouped reference object, and then click Global Object Parameter Values.



2. Specify a value for each parameter. The value can be a tag, backing tag, or numeric or string constant. Numeric constants are treated as strings of text.

   For details about using the Global Object Parameter Values dialog box, see Help.

Using process faceplates

Process faceplates are pre-configured graphic displays and global object displays that interact with Logix5000 processors.

Process faceplates contain graphic objects that display values from a Logix5000 processor and allow operators to interact with the processor. You can use the faceplate graphic displays as stand-alone displays, or copy the faceplate objects into other graphic displays. If desired, you can edit the objects in the faceplates. For example, you might want to add your company logo or change the colors used in the faceplates.

To use a process faceplate, you must add it to your application and specify the path to the Logix5000 processor instructions used by the faceplate. The faceplates are set up so that you can specify the path to the instructions using parameter files. For more information, see the Help for the faceplates, accessible from the Add Process Faceplates dialog box. You can also use global object parameters to specify the path to the instructions.

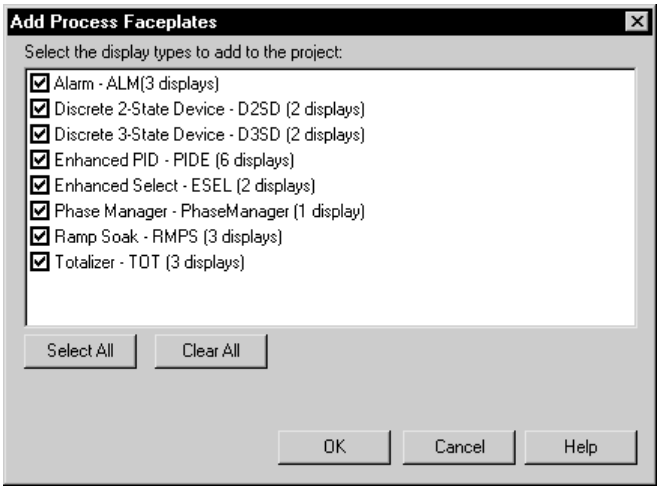
FactoryTalk View Studio comes with sets of faceplates that work with these Logix5000 instructions:

- Enhanced PID (Logix_PIDE)
- Discrete 2-State Device (Logix_D2SD)
- Discrete 3-State Device (Logix_D3SD)
- Totalizer (Logix_TOT)
- Enhanced Select (Logix_ESEL)
- Alarm (Logix_ALM)
- Ramp/Soak (Logix_RMPS)
- PhaseManager (Logix_PhaseManager)

The name in parentheses is the name of the main graphic display and global object display. Other displays in the set are named beginning with this name, followed by a segment describing the purpose of the subdisplay. For example, Logix_ALM has two subdisplays called Logix_ALM_Config and Logix_ALM_Status. When you add process faceplates, any graphic images associated with the faceplates are added to the Images folder.

To add process faceplates to an application

1. In the Explorer window, right-click the HMI server, and then click Add Process Faceplates.


Add Process Faceplates
Select the display types to add to the project:
Alarm - ALM(3 displays)
Discrete 2-State Device - D2SD (2 displays)
Discrete 3-State Device - D3SD (2 displays)
Enhanced PID - PIDE (6 displays)
Enhanced Select - ESEL (2 displays)
Phase Manager - PhaseManager (1 display)
Ramp Soak - RMPS (3 displays)
Totalizer - TOT (3 displays)
Select All
Clear All
OK
Cancel
Help

The Add Process Faceplates dialog box opens.

2. Select the faceplates to add.

   For details about selecting options in the dialog box, see Help.

26 Setting up data logging

This chapter describes:

- steps for setting up data logging.
- data log files.
- data log models.
- changing the data log model used at run time.
- displaying data logs using the trend graphic object.
- problems with data logging.

Summary of steps

Follow these steps to set up data logging:

1. In the Data Log Models editor, set up a data log model that specifies how many log values to store, the conditions that trigger data logging, where to log data, and which tags to monitor.
2. In the Startup editor, turn on data logging by selecting the Data logging check box. Also use this editor to specify the data log model to use at run time. See Chapter 14.
3. In the Graphics editor, create a graphic display containing the trend graphic object, and set up how the object looks, which tag values to display, the start time, and the time span for the data. Also specify the name of the data log model to use.

Data log files

As soon as the application starts running, FactoryTalk® View begins logging tag values to the data log files. When the maximum number of data points have been logged, the oldest data is deleted to make room for the new data. FactoryTalk View supplies data from the log files to the trend object for the requested tags and time span.

The data log files are retained when you restart an application after a shutdown or power loss. You can delete the log files from the runtime computer at application startup.

Use data logging to keep a permanent record of tag data. You can record tag data as tag values change, or on a periodic basis (for example, every minute).

For information about deleting the log files, see page 15-15.

File names

FactoryTalk View creates two data log files:

- *Data Log Model Name*.log
- *Data Log Model Name*.tag

You'll need this information if you log to a custom path and want to delete the log files manually.

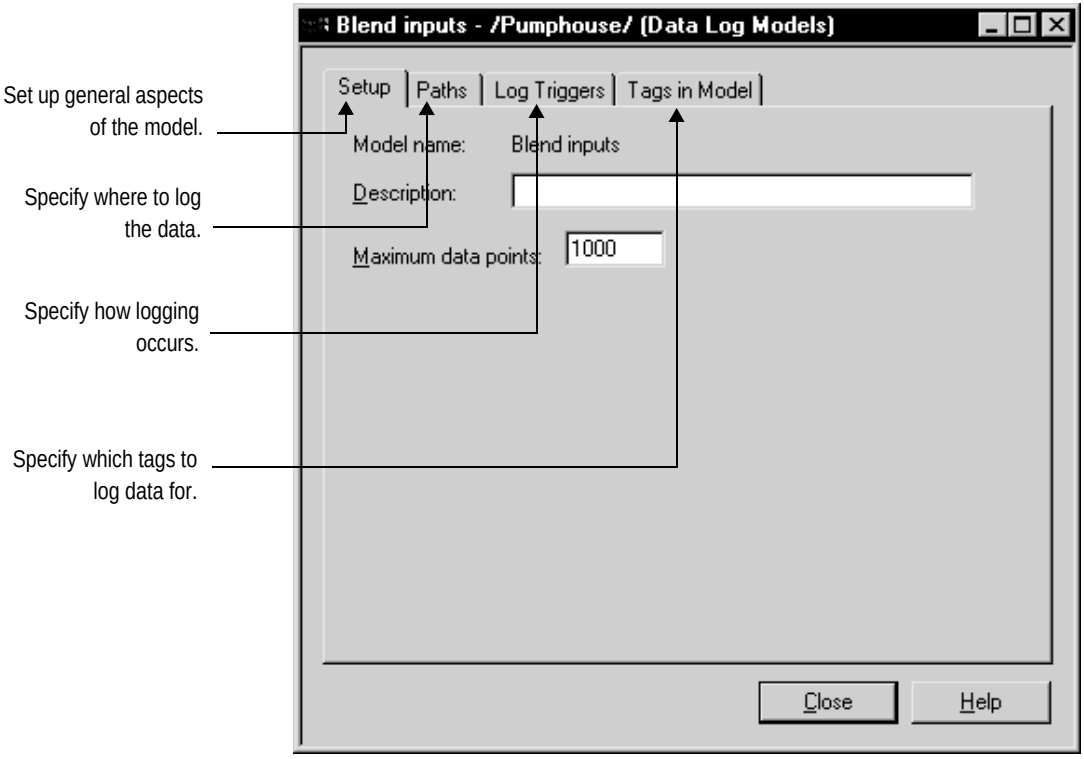
Data Log Models

A data log model defines which tags to log data for, as well as how and where the data is logged.

You can set up multiple data log models, but you can run only one data log model at a time. For information about running a different data log model, see page 26-5.

Creating Data Log Models

Use the Data Log Models editor to create one or more data log models. Each data log model is a file stored in the editor's folder—you can open and work on multiple models at the same time.



For detailed information about the options in the Data Log Models editor, see Help.

Each data log model has a unique name, and an optional description.

You can log a maximum of 300,000 points. When the maximum number of data points have been logged, the oldest data is deleted to make room for the new data.

Data storage locations

You can store data log files in any one of these locations:

- on the runtime computer

- on another computer on the network
- on a compact Flash Card (for PanelView™ Plus or PanelView™ Plus CE terminals only)

If the logging destination is unavailable for any reason, for example, the card is removed or the network drive becomes unavailable, data logging stops. It does not restart until you restart the application.

Logging to a network location

To log data to a network location, the network drive must be shared, and the runtime computer must be logged in to the same domain as the computer on the network. To do this, the user must have access rights for the domain.

The PanelView Plus or PanelView Plus CE terminal cannot be part of a domain. However, you can verify that the user who is logged into the PanelView Plus or PanelView Plus CE terminal is on a user list that is part of a domain.

For more information about logging in to a Windows® domain or authenticating users, see page 17-1.

Data logging methods

There are two methods for triggering data logging. You can set up logging so tag values are logged:

- periodically (periodic logging)
- only when a tag value changes (on-change logging)

Logging periodically

Periodic logging is used to take a snapshot of all tag values at a particular point in time. Tag values are logged even if there has been no change.

You cannot change the periodic log rate at run time.

Logging on change

On-change logging is used to log only tags whose values have changed.

For HMI tags, before logging occurs, the change must equal a specified percentage of change in the tag value. The percentage is based on HMI tags' minimum and maximum values as set up in the Tags editor. Only the tags that change by the specified percentage are logged. If you specify a percentage of 0, all changes are logged.

If a tag does not have a minimum or maximum attribute (for example a data server tag in Logix5000), when you specify on-change logging, all changes are logged for that tag. You don't need to specify a percentage.

Using a heartbeat rate

You can also specify a heartbeat rate, to log values at specified times even if no tag value changes have occurred. The heartbeat ensures that the data in the log file is current. The heartbeat is also a good way to ensure that data logging is working and acquiring valid data.

The heartbeat cannot be less than the maximum update rate, which is the rate at which data servers send tag values to FactoryTalk View.

If you specify a heartbeat of 0, the heartbeat is not used.

Tags in the data log model

The data log model can contain up to 100 analog or digital tags, including both HMI and data server tags.

You cannot use string tags, array tags, tag placeholders, parameters, or expressions in your data log model.

Deleting tags from the model

If you delete a tag from the data log model, previously logged data for the tag is not accessible unless you add the tag back to the model.

Changing the data log model used at run time

Your application can run only one data log model at a time. Follow this procedure to switch data log models.

To run a different data log model

1. Shut down the application.
2. Start FactoryTalk View Studio and open the application.
3. In the Startup editor, specify the new data log model.
4. Create the runtime application. See Chapter 14.
5. Transfer the runtime application to the runtime platform.

   For information about transferring applications to:

   - a personal computer, see Chapter 15.
   - a PanelView Plus or PanelView Plus CE terminal, see Chapter 16.
6. Run the new application.

Displaying data logs using the trend graphic object

You can use the trend graphic object to display the data that you've logged. At run time, when the operator opens a graphic display containing a trend, the trend displays values from the data log file for the data log model that is running. The data log model specifies which data to collect in the data log file.

In addition to displaying historical values from the data log file, trends can display current values for the tags in the model. Trends can also display current values for tags or expressions that are not in a data log model. For more information about trends, see Chapter 28.

Problems with data logging

Problems with data logging occur under these circumstances:

- When your application starts at run time, if any of the tags specified in the current data log model do not exist, an error message is sent to FactoryTalk® Diagnostics.
- If the data log file is corrupted or invalid, the file is deleted and recreated, and an error message is sent to FactoryTalk Diagnostics.
- If logging to a folder on a networked computer, and the runtime computer is not logged in to the Windows domain of the network computer, the log folder cannot be created. An error message is sent to FactoryTalk Diagnostics.
- If logging to an invalid path, the log folder cannot be created. An error message is sent to FactoryTalk Diagnostics. One of the reasons that the path might be invalid is that the top-level folder of the path is not shared.
- When the application starts, FactoryTalk View checks whether there is disk space to store the data log model's data. If there is not enough space, an error message is sent to FactoryTalk Diagnostics and data logging does not start.

27 Using information messages

This chapter describes:

- steps for setting up information messages.
- using the Information Messages and Information Setup editors.
- preparing to set up information messages.
- how information messages work.
- creating information messages in multiple languages.
- the [INFORMATION] display.
- creating your own information message display.
- opening and closing the information message display.
- how the information message display graphic object works.
- changing the information message file used at run time.

About information messages

Use information messages to give the operator messages about the process, prompts or instructions, and information about current states.

Information messages versus local messages

Use information messages to give the operator information no matter which display is open. To give the operator information in a specific graphic display while the display is open, use local messages.

For details about local messages, see page 19-25.

Summary of steps

Follow these steps to set up information messages:

1. In the Information Messages editor, set up the messages and their trigger values.
2. In the Information Setup editor, specify the graphic display to open when information messages occur, and the file of messages to display.

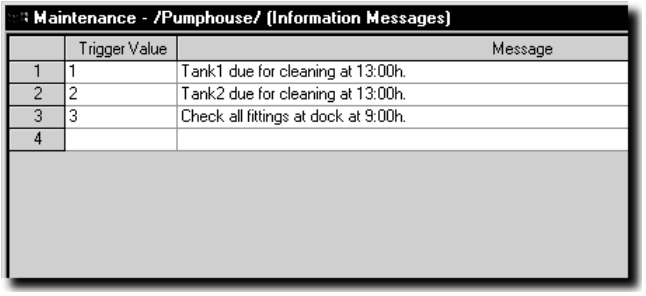
   Also use this editor to assign a tag or expression to the Value connection. If you want the operator to acknowledge messages, assign the Ack connection and specify the acknowledge hold time.

3. In the Startup editor, ensure that the Information messages box is checked (it is checked by default). See Chapter 14.
4. If desired, in the Graphics editor modify the default [INFORMATION] display, or create your own graphic display to use for information messages. For example, if you don't want the operator to acknowledge messages, edit the default display to remove the information acknowledge button.

   For information about graphic displays, see Chapter 19.

Using the Information Messages editor

Use the Information Messages editor to create one or more files of information messages. Each file is stored in the editor's folder. You can open and work on multiple message files at the same time.

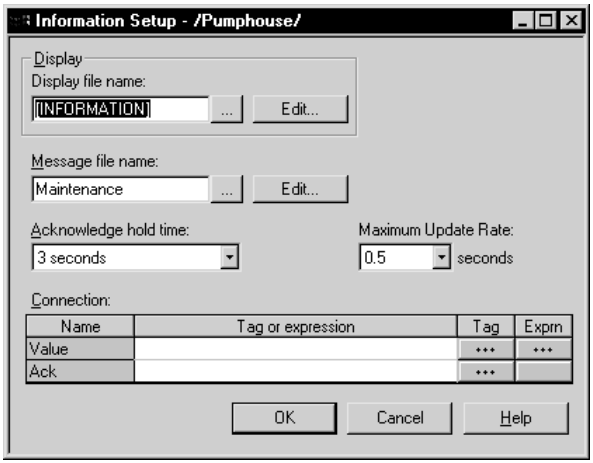


You can define up to 10,000 messages in each message file.

For detailed information about the options in the Information Messages editor, see Help.

Setting up how information messages are displayed

Use the Information Setup editor to set up how information messages are displayed.



For detailed information about the options in the Information Setup editor, see Help.

Preparing to set up information messages

As your application is running, information is continually sent to the data source about the state of various processes. For example, your application might be monitoring whether a valve is open or closed, or the temperature in a boiler. Values representing the status of these processes are sent to the data source.

The data source

The FactoryTalk® View documentation uses the term data source as a generic term that includes all possible sources of tag data, for both data server tags and HMI tags. The data source can be memory or a device such as a programmable controller or an OPC® server. FactoryTalk View writes values to and reads values from the data source. The data source is set up to exchange information (in the form of numeric or string values) between FactoryTalk View and the machine that your application is controlling.

Identifying the tags and values to monitor

To set up information messages, determine which tags associated with machine processes to monitor, and identify the values for those tags that will trigger information messages.

For information about creating tags, see Chapter 7.

How information messages work

These are the key parts of the information message system:

- Information message file—a text file containing a list of messages, with a numeric trigger value for each message
- Information message display—a graphic display that opens at run time and displays information messages
- Value connection—a tag or expression. When the value of this connection matches a message's trigger value, the information message display opens with the associated message displayed.

The following example shows how the key parts of the information message system work together.

Example: Setting up the data source to display information messages

This example shows how to set up the data source to notify the operator that a conveyor belt has stopped. In this example, the status of two conveyor belts is being monitored. You can use both HMI and data server tags.

1. Create a digital tag called Conveyor_1_status. This tag points to an address in a programmable controller that is linked to a sensor on the first conveyor belt. When the belt is running, the tag's value is 0. When the belt stops running, the value changes to 1.
2. Create a second digital tag called Conveyor_2_status. This tag points to an address in a programmable controller that is linked to a sensor on the second conveyor belt. When the belt is running, the tag's value is 0. When the belt stops running, the value changes to 1.
3. Create an analog tag called Information_messages. Set up the data source to send a value of 1 to this tag when Conveyor_1_status has a value of 1, and to send a value of 2 to this tag when the Conveyor_2_status tag has a value of 1.
4. In the Information Messages editor, create these messages with trigger values matching the values that will be sent to the Information_messages tag:

Trigger value	Message
1	Conveyor belt 1 has stopped.
2	Conveyor belt 2 has stopped.

Save the message file with the name “Conveyor belts.”

5. In the Information Setup editor, assign the Information_messages tag to the Value connection, and select the Conveyor belts message file.

At run time, when the value of Conveyor_1_status changes to 1, the first message is displayed (in the default information message display). When the value of Conveyor_2_status changes to 1, the second message is displayed.

Information messages and trigger values

Create messages associated with each tag value change that you want to inform the operator about. Assign each message a trigger value, and set up the data source to send the trigger value to the Value connection. You can use both HMI and data server tags.

The trigger value can be any non-zero integer value (positive or negative). Trigger values do not need to be contiguous, but they must be unique for each message. For example, you could use trigger values of 1, 2, and 3, or values of 10, 20, and 30.

Trigger values cannot be 0. Digital tags have two possible values, 0 and 1. Therefore, if you use a digital tag you can only use the value 1 to trigger a message. If you want to use a digital tag to trigger two different messages, create an expression that adds 1 to the digital tag’s value. That way, you can use the trigger values 1 and 2.

If you use an analog tag or an expression, you can use any non-zero integer or floating-point value to trigger an alarm. Floating-point values are rounded to the nearest integer. For information about how values are rounded, see page 7-2.

You can create multiple information message files, but you can use only one message file at run time. For information about using a different message file, see page 27-8.

Creating information messages in multiple languages

FactoryTalk View 5.00 supports information messages in multiple languages. When you create information messages, they are in the current application language. You can export the information messages for translation and then import them back into the application. For details, see Chapter 12.

Language switching information messages in RSView ME Station 4.00

For applications that will run in RSView ME Station version 4.00, use the CurrentLanguage( ) expression function to specify message offsets in the information message file. In the file, divide your messages into sections for each language. For information about the CurrentLanguage( ) function, see page 23-15.

The [INFORMATION] display

FactoryTalk View comes with an information message graphic display called [INFORMATION]. It contains an information message display graphic object and buttons for acknowledging the displayed message and closing the display. By default, the [INFORMATION] graphic display opens automatically at run time when an information message is generated.



You can use the [INFORMATION] display as is, or modify the display. For example, you can change the color of the objects, or add and remove buttons. Or, you can create your own graphic display to use for displaying information messages. In the Information Setup editor, specify the display to use.

Another option is to place the information message display object in a graphic display that doesn't open automatically when a message is generated.

For information about creating your own information message display, see page 27-7.

The information message display graphic object

The [INFORMATION] display contains an information message display graphic object, which lists one information message at a time.

For information about how the information message display object works at run time, see page 27-7.

Buttons in the [INFORMATION] display

The [INFORMATION] display contains these buttons:

This button	Does this
Ack (information acknowledge)	Acknowledges the information message
Close (close display)	Closes the information message graphic display.

You can assign any caption you choose to the labels on the buttons.

Using the information acknowledge button

When the operator presses the information acknowledge button, if the Ack connection is assigned, the connection value is set to 1 at the data source. The value is held as long as the operator presses the button, or for the acknowledge hold time, whichever is longer. Then the connection is reset to 0.

Creating your own information message display

You can create your own graphic display for displaying information messages, containing an information message display graphic object and the buttons you want to use in the display.

If you create your own graphic display, use an On Top display and select the Cannot Be Replaced option.

For more information about the information message display graphic object, see page 27-7. For information about creating graphic displays and graphic objects, see Chapter 19 and Chapter 20.

Opening and closing the information message display

Opening the display

The information message display you specify in the Information Setup editor (either the default [INFORMATION] display or your own display) is automatically opened whenever the Value connection's value matches a trigger value.

You can also create a goto display button that the operator can press to open the information message display. For information about setting up a goto display button and specifying the display to open, see Help.

Closing the display

The information message display closes when the Value connection's value changes to 0.

To automatically close the display when the operator acknowledges a message, set up the data source to set the Value connection to 0 when the Ack connection is set to 1.

The operator can also close the display by pressing a close display button.

How the information message display graphic object works

At run time, when the Value connection at the data source changes from 0 to a non-zero value, the assigned information message display opens. If the value matches a message's trigger value, the associated information message appears in the information message display graphic object. The object can be in the default [INFORMATION] display, in an

information message display you have created, or can be placed in any display in your application.

For information about setting up the information message display graphic object, see page 21-63.

What is displayed

- If the Value connection is unassigned, the information message graphic display never opens automatically. If the operator opens a graphic display containing an information message display object, the object is blank.
- The Value connection is rounded up to the nearest integer. If the value does not match any of the trigger values in the specified message file, the display is filled with question marks (?).

  For information about how values are rounded, see page 7-2.
- If the message is too long to fit in the information message display object, the last displayed character is replaced with an asterisk (*).
- When the Value connection's value is 0, the information message graphic display is closed.
- If you set up information messages in multiple languages, messages are displayed in the current application language. When a language switch occurs, a message that was already in the information message display remains in the language that it originally appeared in. New messages are displayed in the new language.

Changing the message file used at run time

Your application can use only one message file at a time. Follow this procedure to switch message files.

To use a different message file

1. Shut down the application.
2. Start FactoryTalk View Studio and open the application.
3. In the Information Setup editor, specify the new information message file.
4. Create the runtime application. See Chapter 14.
5. Transfer the runtime application to the runtime platform.

   For information about transferring applications to:

   - a personal computer, see Chapter 15.
   - a PanelView™ Plus or PanelView™ Plus CE terminal, see Chapter 16.

6. Run the new application.

28 Setting up trends

This chapter describes:

- what trends are.
- summary of steps for creating a trend graphic object.
- creating trends, and the Trend Object Properties dialog box.
- the parts of the trend graphic object.
- the different chart types.
- choosing colors, fonts, lines, and markers for the trend.
- testing the trend.
- using objects from the Trends graphic library.
- using buttons to control the trend at run time.
- printing trend data.
- runtime errors for trends.

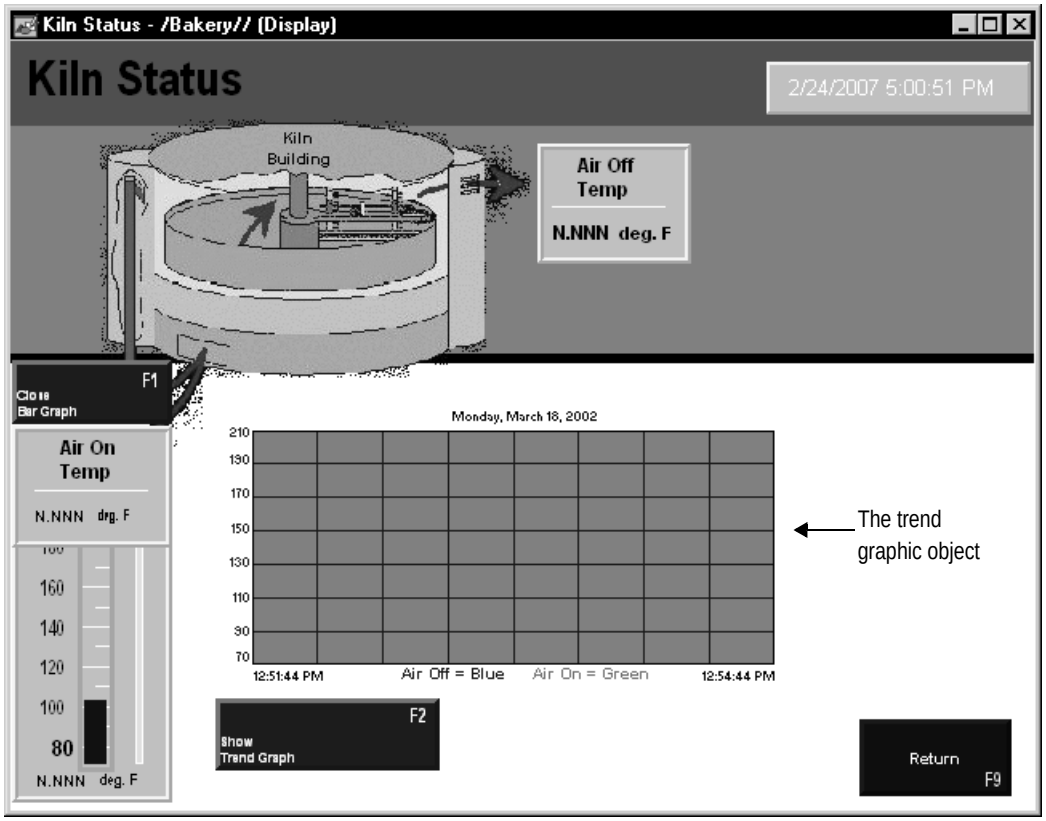
About trends

A trend is a visual representation of current or historical tag values. The trend provides operators with a way of tracking plant activity as it is happening.

You can:

- plot data for as many as eight tags or expressions on one trend.
- create a trend that is part of a graphic display or acts as the entire graphic display.
- plot data over time, or plot one variable against another in an XY Plot chart to show the relationship between them.
- display isolated or non-isolated graphs. Isolated graphing places each pen in a separate band of the chart. With non-isolated graphing, pen values can overlap.
- create buttons to allow the operator to pause, scroll, and print the trend data.

The illustration below shows a trend that has been added to a graphic display. You can view the Kiln Status display by opening the Malthouse sample application.



Current versus historical data

The data displayed in a trend can come from two sources. For current values, data comes from the value table as it is collected. The value table is a record of the most recent values collected from the data source, and is stored in temporary memory while the application is running.

For historical values, data comes from a data log model's log file, if a model is assigned to the trend. You can display both current and historical data in the same trend.

For information about data log models, see Chapter 26.

Time, date, and number formats

The trend is displayed using the time, date, and number formats of the current application language. For example, if the application language uses a comma for the decimal symbol, the scale on the y-axis uses commas for the decimal symbol.

For information about using multiple languages, see Chapter 12.

Summary of steps

These are the steps for creating a trend:

1. To plot historical data, create a data log model in the Data Log Models editor. For information, see Chapter 26.
2. Create a trend graphic object in the Graphics editor, as described on page 28-3.
3. Set up the trend in the Trend Object Properties dialog box. For details about the options in the dialog box, see Help.
4. If desired, create a next pen button, a pause button, or key buttons in the same graphic display, to allow the operator to switch between pens, pause the trend, or scroll the trend.

   For information about the buttons you can use with trends, see page 28-11.
5. To keep a printed record of the trend data, provide a way for the operator to print the graphic display. For information see page 28-12.

Creating trend objects

To create a trend object

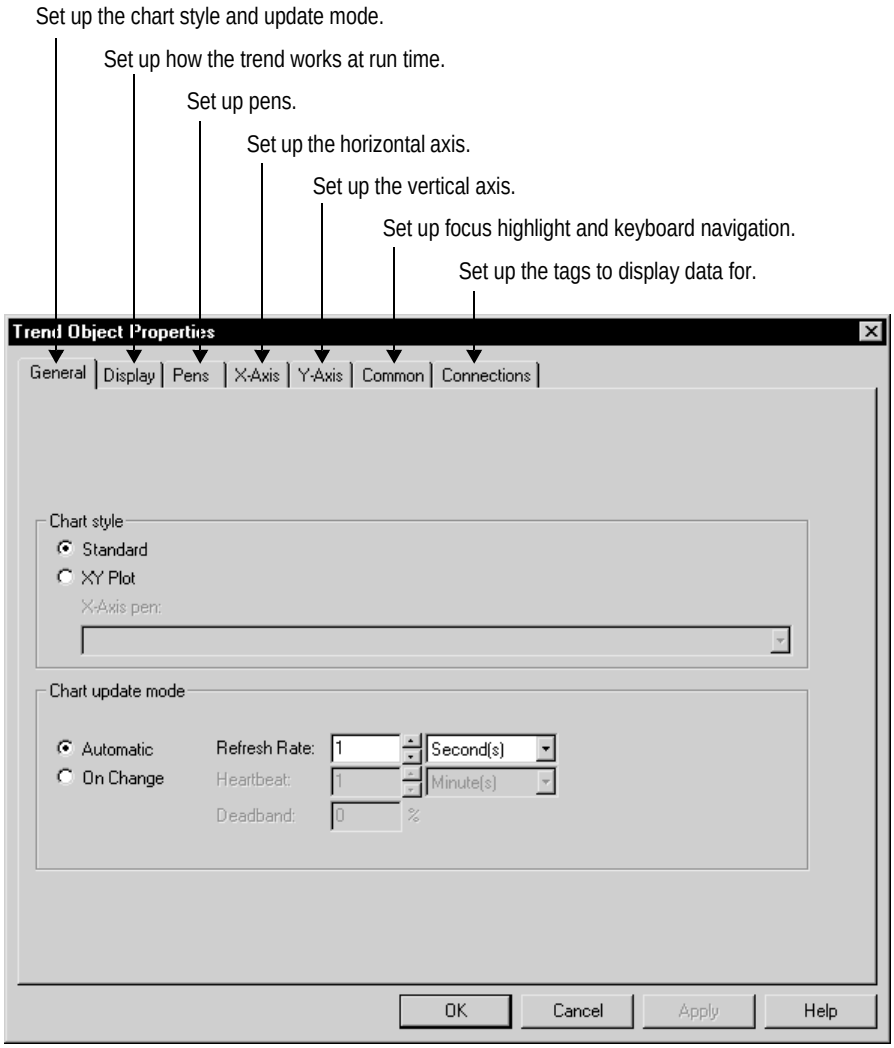
1. In the Graphics editor, create or open a graphic display.
2. Select the Trend drawing tool by doing one of the following:
   - In the Objects toolbox, click the Trend tool.
   - On the Objects menu, select Trending, and then click Trend.
3. Drag the mouse to create a box approximately the size you want for the trend.
4. Double-click the trend to open the Trend Object Properties dialog box.
5. Set up the trend. For details, see Help.

Once you have set up the trend, you can edit it as you would any other graphic object. You can move it, resize it, attach animation to it, and so on. You can also use this object in other graphic displays by dragging it from one display and dropping it into another.

For more information about graphic objects, see Chapter 20.

Setting up trends

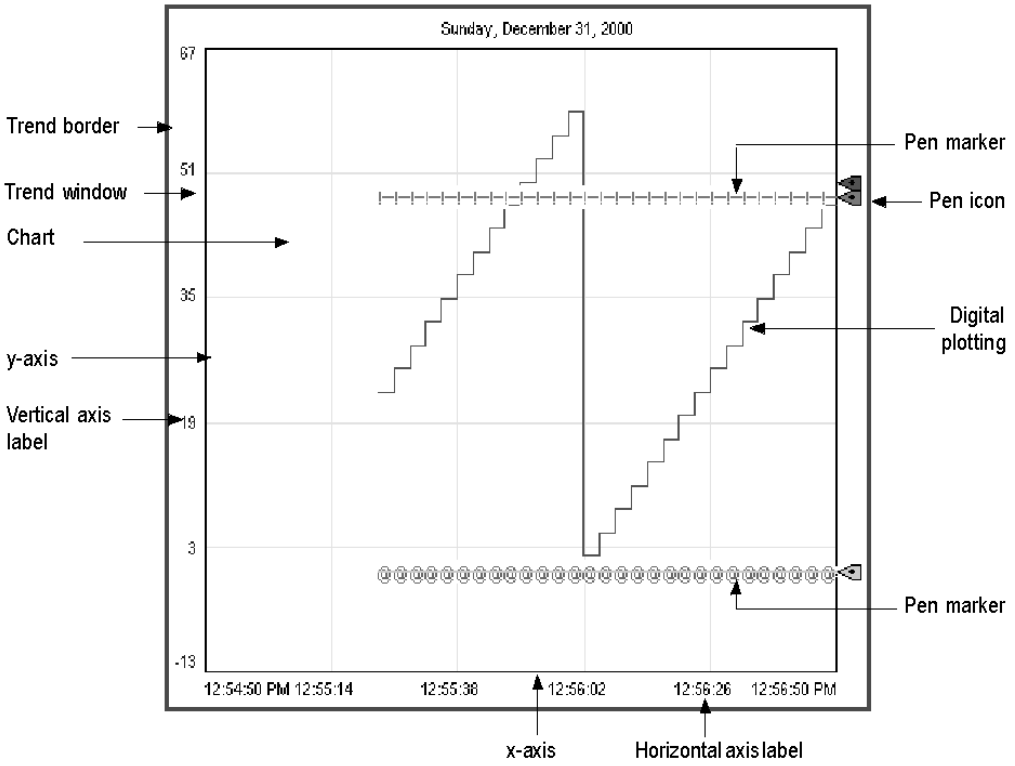
When you double-click a trend object, the Trend Object Properties dialog box opens. Use the dialog box to set up the trend.



For details about the options in the Trend Object Properties dialog box, see Help.

The parts of a trend

The illustration below shows a standard trend chart, with three pens and a two-minute time span. Two of the pens have markers. The third uses digital plotting. For more information about chart types, see page 28-7.



Trend border

The border appears around the trend object at run time when the trend is selected.

Trend window

The area around the chart, between the border and the chart, is the trend window.

Chart

The chart is the area of the trend in which values are plotted. It is bounded by the y-axis on the left and the x-axis on the bottom. It contains the plotted trend data (shown using pen lines and pen markers), as well as grid lines (if you choose to display them).

Y-axis

The y-axis is the left vertical edge of the chart. It is also known as the vertical axis.

Vertical axis labels

The vertical axis labels show the scale (range) of values for the pens. If desired, you can set up the trend to omit the vertical axis labels.

The minimum and maximum values for the scale can be determined automatically (using the best fit for the current data), be derived from a pen's minimum and maximum values, use a constant value, or be controlled by tags.

You can set up the trend so all pens use the same scale, or use individual ranges for each pen. If you choose the latter method, create a next pen button in the graphic display, to allow operators to view the range for each pen. When the operator presses the button, the vertical axis changes to the new pen's range.

For example, if Pen 1 has a minimum value of 10 and a maximum value of 100, the range on the vertical axis is 10 to 100 when the pen is selected. If Pen 2 has a minimum of -10 and a maximum of 50, the range on the vertical axis changes to -10 to 50 when the operator presses the next pen button.

X-axis

The x-axis is the bottom horizontal edge of the chart. It is also known as the horizontal axis.

Horizontal axis labels

For standard charts, the horizontal axis labels indicate the time span covered by the trend. For XY Plot charts, the horizontal axis labels show the scale (range) of values for the pen selected to serve as the x-axis pen.

If desired, you can set up the trend to omit the horizontal axis labels. The number of labels depends on the size of the trend object and the number of vertical grid lines.

Pens

Pens are the lines and symbols used to represent values. The values can be tags you are monitoring, expressions that manipulate tag values, or constants.

If there is no data for a pen, or if the data is outside the vertical axis range, the pen does not appear in the chart.

Pen icons

Pen icons appear at the right edge of the chart at run time, if you choose to display them. The icon's position indicates the pen's most recent recorded value (from the value table), even if the trend is paused or if the most recent value has not been plotted yet.

Pen markers

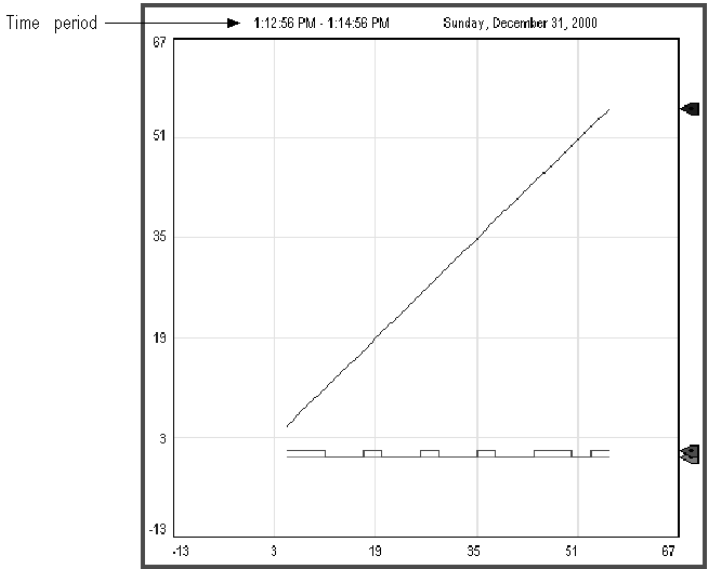
Pen markers are symbols that indicate data points. If data is plotted frequently, the markers might not appear as distinct, separate symbols. For example, see the lowest pen in the illustration on page 28-5.

Chart types

Standard vs. XY Plots

You can create a standard chart, which plots tag values against time, or an XY Plot chart, which plots one (or more) tag's values against another's.

This illustration shows what an XY Plot chart could look like:

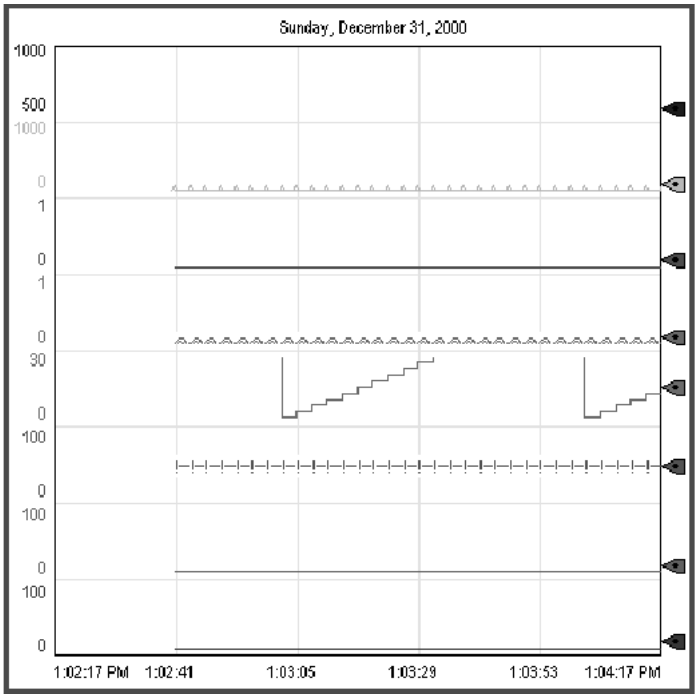


Notice that the horizontal axis labels display the range for the specified x-axis pen. The time period covered by the chart is at the upper left.

Isolated graphing

For charts with multiple pens, you can allow the pen values to overlap, or you can isolate each pen in its own horizontal band on the chart.

This is an example of isolated graphing, with a 10% buffer between each pen's band:



Notice that in this illustration each pen uses its own scale. If desired, you can use the same scale for all pens.

With isolated graphing, a grid line is automatically placed above each pen's band.

Plotting a value across the full width of the chart

Use horizontal lines to provide a frame of reference for your tag data. For example, if you define values that are the limits within which a tag must operate, and display horizontal lines in your trend to indicate the limits, when a tag crosses one of these limits the tag's alarm condition is obvious on the trend.

There are two ways to plot a value across the full width of the chart:

- In the Connections tab, assign a constant value to a pen.

When values for the pen have been plotted across the full width of the chart, the pen appears as a solid line.

- In the Connections tab, assign to a pen the tag, expression, or constant whose value will be used to determine the position of the line, and then in the Pens tab, choose the pen type Full Width.

  As soon as the trend is displayed, the pen appears as a horizontal line across the full width of the chart. Its vertical position is determined by the tag, expression, or constant's value. If the value changes, the position changes.

Choosing trend colors, fonts, lines, and markers

The following table summarizes where in the Trend Object Properties dialog box to specify colors, fonts, lines, and markers for a trend.

You can also specify these settings in the Properties tab of the Property Panel.

To specify this	Use this box or column	In this tab
Chart background color	Background color	Display
Horizontal label color	Text color	Display
Text font, style, and size	Font (button)	Display
Pen line, pen marker, pen icon, and vertical label color	Color	Pens
Pen line width	Width	Pens
Pen line style	Style	Pens
Pen marker	Marker	Pens
Vertical grid line color	Grid color	X-Axis
Horizontal grid line color	Grid color	Y-Axis

The trend border color

The trend border uses the highlight color for the graphic display, specified in the Behavior tab of the Display Settings dialog box.

The trend window color

By default, the trend window uses the background color of the display, specified in the General tab of the Display Settings dialog box.

To use a different window color

1. In the Property Panel, select the opaque WindowStyle, and then specify the WindowColor property.

For information about using the Property Panel, see page 20-29.

Testing the trend



Test Display tool

You can quickly test the trend by switching to test mode. If communications are active and there is data for the tags, the pens plot values in the trend. When you are finished testing, switch back to edit mode to continue editing.



Edit Display tool

To switch between test and edit modes

1. On the View menu, click Test Display or Edit Display, or click the Test Display and Edit Display tools.



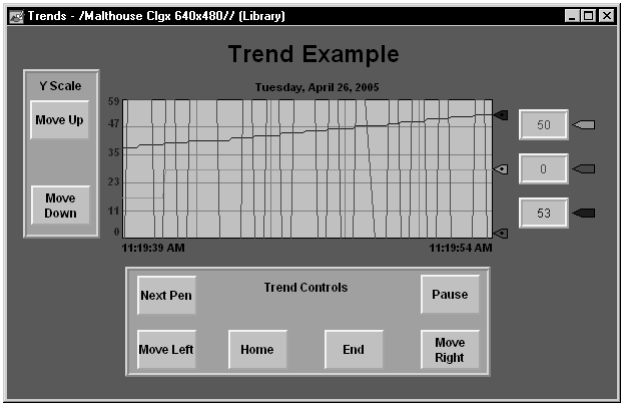
Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the Display Settings dialog box. Also, data logging is not turned on in test mode.

Using the Trends graphic library

The Trends graphic library contains a trend graphic object and buttons for controlling the trend. It also contains numeric display objects that display the value of each tag used in the trend.

You can use the trend and objects as they are, or you can edit them to suit your needs. To use the objects, drag and drop (or copy and paste) them into your graphic display.

For information about copying and pasting objects from the graphic libraries, see page 20-43.



To use the Trends graphic library

1. Open the Graphics folder, and then open the Libraries folder.
2. Double-click the Trends library.
3. Drag and drop or copy and paste objects into your display.

Using buttons to control the trend at run time

You can use button graphic objects with the trend, to allow the operator to pause the trend, switch between pens, or scroll the trend.

You can link buttons to a specific trend object, or set up a button to work with whichever object is selected in the graphic display. For information about linking buttons to objects, see page 21-9.

Use these buttons with trends:

This button	Does this
Pause	Toggles between pausing and automatic scrolling. When the trend is paused, the pen icons continue to move vertically to indicate the pens' current values. When the trend resumes scrolling, values that occurred while the trend was paused are filled in, bringing the trend up to the current time (unless you are scrolling historical data).
Next pen	Changes the vertical axis labels to the scale for the next pen. The color of the labels matches the color of the selected pen.
Move up	Scrolls up to display higher values on the vertical scale. For example, if the visible scale range is 0 to 100, pressing move up could change the visible range to 10 to 110. The incremental amount the axis scrolls depends on the pen's range and the number of horizontal grid lines. This button does not work if the "Minimum / maximum value option" in the Y-Axis tab is set to Automatic.
Move down	Scrolls down to display lower values on the vertical scale. This button does not work if the "Minimum /maximum value option" in the Y-Axis tab is set to Automatic.
Move left	Pauses the trend and scrolls to the left.
Move right	Pauses the trend and scrolls to the right.
Home	Pauses the trend and moves to the earliest data in the trend.

This button	Does this
End	Resumes trend scrolling and moves to the current (latest) data in the trend.

To see how the buttons work with the trend, open the Trends graphic library (see page 28-10), and start test mode.

For information about creating buttons, see Chapter 20. For details about setting up the buttons, see page 21-16.

Printing trend data

To print trend data at run time, provide the operator with a method for printing the graphic display.

You can use these methods to print graphic displays at run time:

- Create a display print button. For information about creating graphic objects, see Chapter 20.
- Assign a tag or expression to the Remote Display Print connection (in the Global Connections editor). When the value of the tag or expression changes from 0 to a non-zero value, the current display is automatically printed.

  Program the data source to trigger the change as often as you want the data printed.

  For more information about setting up remote display printing, see Chapter 8.

Everything on the screen is printed, including the current display, pop-up windows, and any visible background applications.

For information about specifying which printer to use at run time for applications that will run on a personal computer, see page 15-10.

For information about specifying printer options for applications that will run on a PanelView™ Plus or PanelView Plus CE terminal, see the *PanelView Plus Terminals User Manual*.

Improving clarity of the trend printout

Depending on what type of printer you use, pen lines with a width of 1 pixel might not appear in the printout. Choose high-contrast colors and wider line widths to ensure that the trend data prints clearly.

Runtime errors for the trend

If data for the trend is not available at run time due to communication errors, a message is sent to FactoryTalk® Diagnostics.

See Help for information about solving common trend problems.

29 Setting up RecipePlus

This chapter describes:

- what recipes are.
- summary of steps for creating a recipe system.
- how the recipe system works.
- specifying the runtime location of recipe files.
- creating recipe files.
- comparing recipes.
- creating RecipePlus buttons, selectors, and tables.
- testing RecipePlus objects.
- using objects from the RecipePlus_Components graphic library.
- using buttons with the recipe objects.
- viewing data values that are saved at run time.

About recipes

A recipe is a set of numeric and string data values (ingredients) that can be downloaded to their associated tags at the data source. Each ingredient has a pre-set data value assigned to it. The set of data values for all the ingredients in a recipe is called a data set. The set of numeric and string tags assigned to the ingredients in the recipe is called a tag set. The ingredients, data sets, and tag sets are stored together in a recipe file.

You can create different pairs of data sets and tag sets for the same set of ingredients. Each pairing of data set with tag set is called a unit. Each unit is like a unique recipe. At run time, the operator can select the unit (recipe) that applies to the current operation.

For example, a bakery making whole wheat bread could use the same ingredients and tag sets, but depending on the type of crust desired, could use different data sets to specify different baking temperatures. As another example, you might want to have multiple production lines baking the same bread. In this case, the data set for all the production lines would be the same, but the tags receiving the recipe information would be different for each production line. Units allow you to combine different tag sets and data sets for the same set of ingredients.

The FactoryTalk® View RecipePlus system allows you to create up to 15,000 ingredients, 500 data sets, 50 tag sets, and 2,500 units for each recipe file. You can create data sets at

development time, edit them at run time, and also create new data sets from tag values at run time. You can write the data set values to tags, or write tag values to data sets.

The RecipePlus system can be used for manufacturing food and beverages, but it can also be used for any application where you want to display, edit, download, or save multiple values at once. For example, recipes are used in the petrochemical and pharmaceutical industries. In the pharmaceutical industry, you could use recipes to design flexible packaging, creating recipes that specify the number of tissues to put in a box or the number of milliliters of shampoo to put in a bottle.

Summary of steps

These are the steps for creating a recipe system:

1. In the RecipePlus Setup editor, specify the runtime location for recipe files. The files can be stored with the application or in a separate location. For details, see page 29-5.
2. In the RecipePlus Editor, set up ingredients, data sets, tag sets, and units. You can also specify a percent complete tag and a status tag for the recipe. For details, see page 29-6.
3. Create a display in the Graphics editor, containing a RecipePlus selector, table, and buttons. For details, see page 29-9.
4. If desired, create key buttons in the same graphic display, to allow the operator to use the selector and table without a keyboard.

   For information about the buttons you can use with recipes, see page 29-11.

How the recipe system works

A recipe system consists of a recipe file and the graphic objects used to work with the ingredients at run time.

RecipePlus selector

Use the RecipePlus selector to select the recipe file and unit to work with.

RecipePlus table

Use the RecipePlus table to display the selected recipe file's ingredients, tag values, and data set values. The operator can edit data set values in the table, unless you select the View only option.

If desired, you can include a Compare column in the table, to compare tag values to data sets at a glance. If you choose this option, FactoryTalk View displays an X in the Compare column when the tag value and data set value for an ingredient differ. Ingredients with an X are listed first.

RecipePlus button

Use the RecipePlus button to perform actions on the selected recipe's ingredients. The recipe is selected using the RecipePlus selector object. Set up a separate RecipePlus button for each action you want to perform:

- Download—write the data set values to tags, for all the ingredients in the selected recipe.
- Upload—write tag values to the data set, for all the ingredients in the selected recipe. If all values are uploaded successfully, the recipe file is saved.
- Upload and Create—write tag values for all the ingredients in the selected recipe to a new data set, creating a new unit. The operator is prompted for a name for the new unit. If all values are uploaded successfully, the recipe file is saved.

  The new data set is named Data Set *n*, where *n* is the next available number (starting at 1) that will create a unique data set name.
- Restore—display the selected recipe in the RecipePlus table.
- Save—save the data set values for the recipe file and unit displayed in the RecipePlus table. If the operator made changes in the data set values using the string pop-up keyboard or numeric pop-up keypad, the new values in the table overwrite existing data set values (if any) for the unit in the recipe file.
- Delete—delete the recipe unit selected in the RecipePlus selector object.
- Rename—rename the recipe unit selected in the RecipePlus selector object.

The illustration below shows a graphic library display that contains a RecipePlus selector, RecipePlus buttons, and a RecipePlus table. The display also contains key buttons for

working with the selector and table, a bar graph that shows the percentage complete of the recipe operation, and a string display that shows the status of the recipe operation.



Number format

The values in the recipe table are displayed using the number format of the current application language. For example, if the application language uses a comma for the decimal symbol, floating-point values in the table use commas for the decimal symbol. For information about using multiple languages, see Chapter 12.

Numeric limits

RecipePlus supports the range of numbers allowed by the VARIANT data type. This range is -1.797693E+308 to 1.797693E+308. This range applies to the numbers that you enter in the RecipePlus Editor, and it also applies to the tag values that are uploaded to the recipe file at run time.

Specifying the runtime file location

Use the RecipePlus Setup editor to specify the runtime file location.

Storing files outside the HMI project

If you store the recipe files outside the HMI project, the runtime application can use updated recipe files without creating a new runtime application (.mer) file.

Storing recipe files outside the HMI project also allows you to use FactoryTalk View Studio to view and edit recipe data that is saved at run time without converting the .mer file to an .med file. For more information, see page 29-13.

If you want to store recipe files outside of the HMI project at run time, make sure you move the files from the application's RecipePlus folder to the specified runtime location before running the application.

This is the path to the RecipePlus folder:

\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects*Project name*\RecipePlus (Windows® 2000)

or

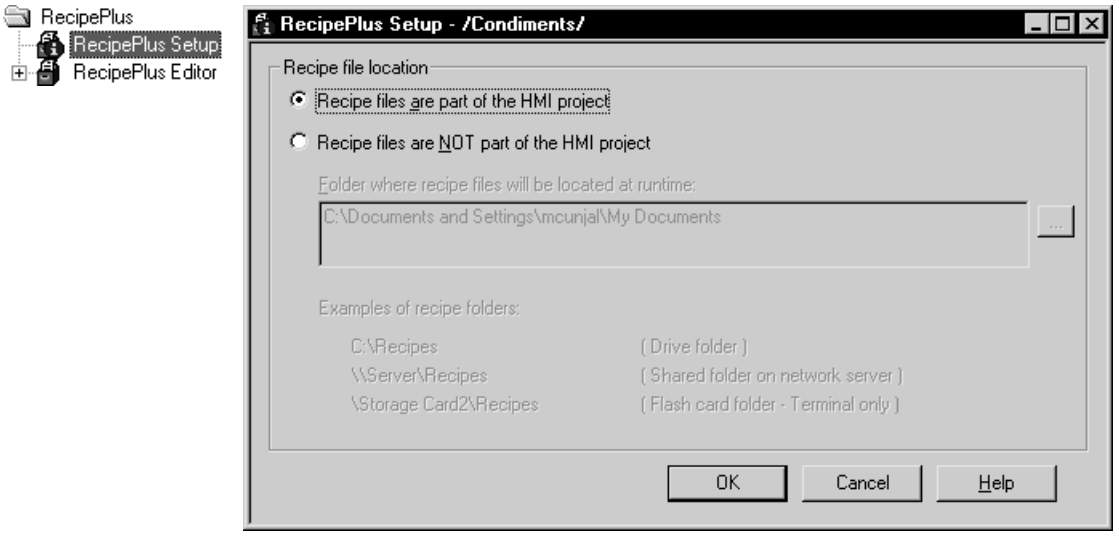
\Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME\HMI projects*Project name*\RecipePlus (Windows XP or Windows Server 2003 R2)

If recipe files are stored outside the HMI project, when you perform an action on a recipe file at run time, the file is locked until the action is completed. This prevents other users from making changes to a file while you are working with it.

Storing recipe files with the HMI project

If recipe files are part of the HMI project, when a recipe file is saved at run time, FactoryTalk View updates the .mer file with changes to the data sets. When you stop the runtime application, the changes are retained, and are displayed the next time you run the application and display the recipe file. You can convert the .mer file to an .med file to

view the changes in FactoryTalk View Studio. For more information about converting runtime application files, see page 14-6.


RecipePlus
RecipePlus Setup
RecipePlus Editor
RecipePlus Setup - /Condiments/
Recipe file location
Recipe files are part of the HMI project
Recipe files are NOT part of the HMI project
Folder where recipe files will be located at runtime:
C:\Documents and Settings\mcunjal\My Documents
Examples of recipe folders:
C:\Recipes (Drive folder)
\\Server\Recipes (Shared folder on network server)
\Storage Card2\Recipes (Flash card folder - Terminal only)
OK
Cancel
Help

For details about using the RecipePlus Setup editor, see Help.

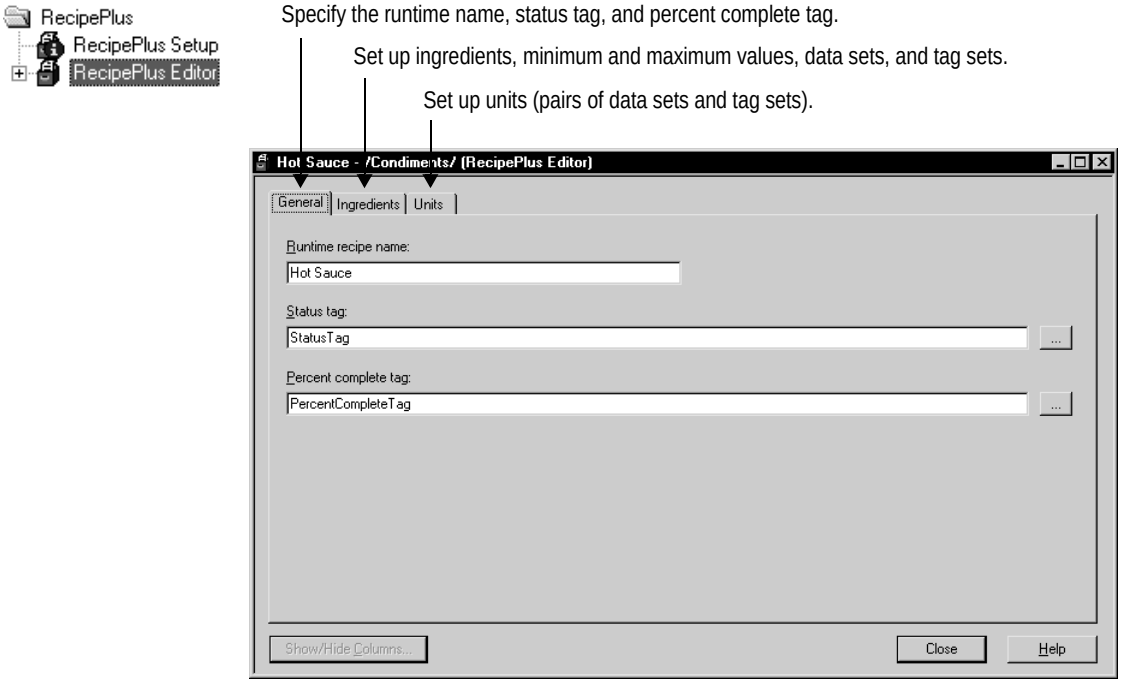
Setting up recipe files

Use the RecipePlus Editor to set up one or more recipe files. Each file is stored in the editor's folder. You can open and work on multiple recipe files at the same time.

The RecipePlus Editor has special items on the Edit menu that allow you to easily copy and paste from the spreadsheet in the Ingredients tab to Microsoft® Excel. This editor also has items on the Recipe menu, for adding, deleting, and renaming data sets and tag sets, and for comparing recipes.

To help you get started, FactoryTalk View creates one data set, tag set, and unit. You can rename them and assign data values and tags to them, or delete them and create your own.

For information about comparing recipes, see the next section.



For details about the options in the RecipePlus Editor, see Help.

You can also use the RecipePlus Editor to view the data values that are saved at run time. For more information, see page 29-13.

Comparing recipes

You can use the RecipePlus Editor to compare data sets and tags sets within a single recipe, or between two recipes.

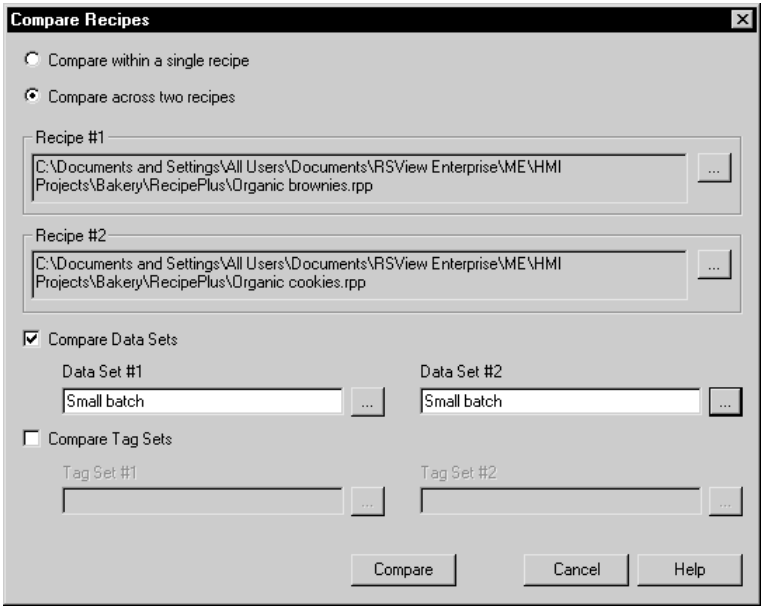
If you are comparing data sets or tag sets within a recipe, only ingredients with different values are displayed in the report.

If you are comparing data sets or tag sets between two recipes, both common ingredients and unique ingredients are listed in the report.

To compare recipes

1. Open the RecipePlus Editor.

2. On the Recipe menu, click Compare Recipes.

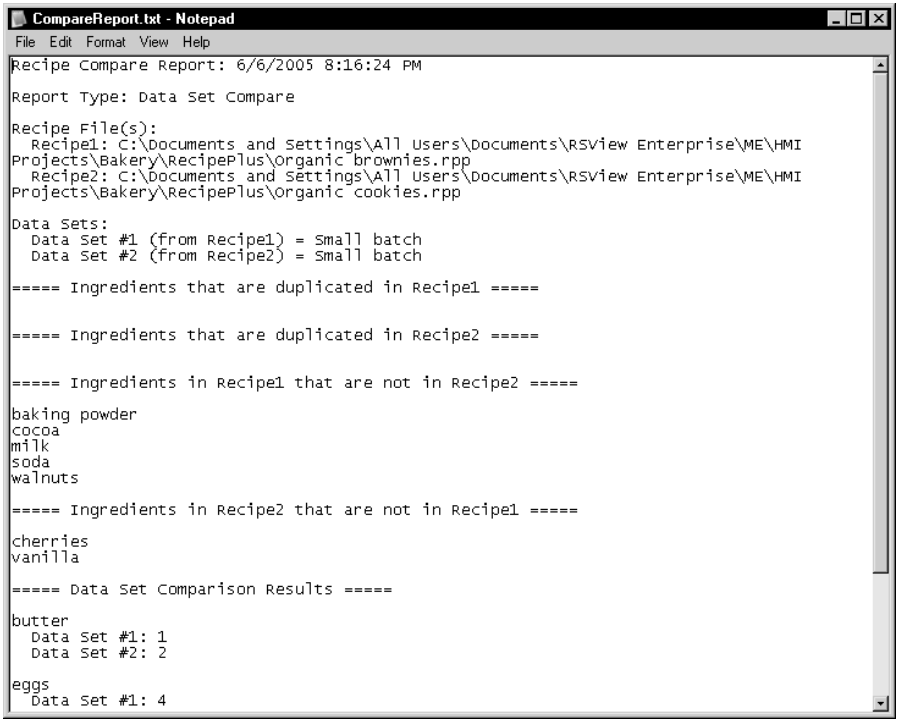


3. Specify the recipe file or files, data sets, and tag sets to compare.

   For information about the options in the Compare Recipes dialog box, see Help.

4. Click Compare.

   A report is displayed in Windows® Notepad.


CompareReport.txt - Notepad
File Edit Format View Help
Recipe Compare Report: 6/6/2005 8:16:24 PM
Report Type: Data Set Compare
Recipe File(s):
Recipe1: C:\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI Projects\Bakery\RecipePlus\Organic brownies.rpp
Recipe2: C:\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI Projects\Bakery\RecipePlus\Organic cookies.rpp
Data Sets:
Data Set #1 (from Recipe1) = Small batch
Data Set #2 (from Recipe2) = Small batch
===== Ingredients that are duplicated in Recipe1 =====
===== Ingredients that are duplicated in Recipe2 =====
===== Ingredients in Recipe1 that are not in Recipe2 =====
baking powder
cocoa
milk
soda
walnuts
===== Ingredients in Recipe2 that are not in Recipe1 =====
cherries
vanilla
===== Data Set Comparison Results =====
butter
Data Set #1: 1
Data Set #2: 2
eggs
Data Set #1: 4

Time and date formats

The time and date in the report use the time and short date format for the current application language. For information about using multiple languages, see Chapter 12.

Printing recipes

You can use the RecipePlus Editor to print recipe data sets. For information about printing from editors, see page 2-13.

Creating RecipePlus objects

You can create one RecipePlus table and RecipePlus selector per graphic display. You can create multiple RecipePlus buttons in a display, with a different action assigned to each.

The objects and button actions to use depend on how you want to use your recipe system. For example, if you just want to write data set values to tags, all you need is a RecipePlus selector and a RecipePlus button with the download action. For information about how the different objects in the recipe system work, see page 29-2.

To create a recipe object

1. In the Graphics editor, create or open a graphic display.
2. Select a RecipePlus drawing tool by doing one of the following:
   - In the Objects toolbox, click the RecipePlus Button, RecipePlus Selector, or RecipePlus Table tool.
   - On the Objects menu, select RecipePlus, and then click RecipePlus Button, RecipePlus Selector, or RecipePlus Table.
3. Drag the mouse to create a box approximately the size you want for the object.
4. Double-click the object to open its Properties dialog box.
5. Set up the object. For details, see Help.

Once you have set up a RecipePlus object, you can edit it as you would any other graphic object. You can move it, resize it, attach animation to it, and so on. You can also use the object in other graphic displays by dragging it from one display and dropping it into another.

For more information about graphic objects, see Chapter 20.

Testing RecipePlus objects



Test Display tool



Edit Display tool

You can quickly test the recipe objects in a display by switching to test mode. If communications are active and there is data for the tags, you can download and upload recipe tag values. When you are finished testing, switch back to edit mode to continue editing.

To switch between test and edit modes

1. On the View menu, click Test Display or Edit Display, or click the Test Display and Edit Display tools.



Test mode is not the same as running the display. Test mode does not change the appearance or position of the display as set up in the Display Settings dialog box.

Using the RecipePlus_Components graphic library

The RecipePlus_Components graphic library contains a RecipePlus selector and table and buttons for working with the objects. It also contains a bar graph and multistate indicator that display the status of recipe operations.

Use test mode to see how the different RecipePlus objects work together. In test mode, the RecipePlus selector in the library displays any recipe files and units that you have created in your application.

You can use the objects in the library as they are, or you can edit them to suit your needs. To use the objects, drag and drop (or copy and paste) them into your graphic display.

For information about copying and pasting objects from the graphic libraries, see page 20-43.



To use the RecipePlus_Components graphic library

1. Open the Graphics folder, and then open the Libraries folder.
2. Double-click the RecipePlus_Components library.
3. Drag and drop or copy and paste objects into your display.

Using buttons with recipe objects

You can use button graphic objects with the RecipePlus selector and table, to select the recipe and unit to work with, and to select ingredients in the table.

You can link buttons to a specific recipe object, or set up a button to work with whichever object is selected in the graphic display. For information about linking buttons to objects, see page 21-9.

Use these buttons with recipe objects:

This button	Does this
Move up	Moves the highlight bar up one item in the list.
Move down	Moves the highlight bar down one item in the list.
Page up	Moves the highlight bar up one page in the list.
Page down	Moves the highlight bar down one page in the list.
Home	Moves the highlight bar to the top item in the list.
End	Moves the highlight bar to the bottom item in the list.
Enter (table only)	Opens the numeric keypad or string keyboard for the operator to edit the data set value. If a numeric ingredient has a minimum and maximum value defined, these values are displayed in the numeric keypad. If the table is defined as View only, the operator cannot edit it.

To see how the buttons work with the RecipePlus selector and table, open the RecipePlus_Components graphic library (see page 29-10), and start test mode.

For information about creating buttons, see Chapter 20. For details about setting up the buttons, see page 21-16.

Example: Editing and downloading recipe values at run time

This example shows how to use the RecipePlus graphic objects to edit and download recipe values at run time.

1. In the RecipePlus editor, create a RecipePlus file containing ingredients, several data sets, a tag set, and several units combining the different data sets with the tag set.
2. Open the RecipePlus_Components library.
3. Start test mode.
4. Use the move up and move down buttons next to the RecipePlus selector to highlight a unit in the selector, and then press the Restore button.

   The unit's ingredients are displayed in the RecipePlus table, with the data set values in the Recipe column.

5. Use the move up and move down buttons next to the RecipePlus table to select an ingredient, and then press the Enter button.

   The numeric pop-up keypad opens, displaying the minimum and maximum values for the ingredient. If the ingredient is a string ingredient, the string pop-up keyboard opens.

6. Type a new value for the ingredient, and then press Enter.

   The new value is displayed in the Recipe column.

7. Press the Save button to save the new value.

8. Press the Download button to write all the values in the Recipe column to the tags associated with the ingredients.

   The values are downloaded to the data source.

Viewing data values that are saved at run time

Use the RecipePlus Editor in FactoryTalk View Studio to view data values that have been saved at run time.

The operator can save tag values at run time by uploading to an existing data set or to a new data set. The operator can also edit data set values in the RecipePlus table and save the edited values (unless the table is View only).

If recipe files are stored with the HMI project, changes are saved in the .mer file. To view the changes in FactoryTalk View Studio, convert the .mer file to an .med file. For more information about converting the runtime application file, see page 14-6.

To view data values in modified recipe files

1. If recipe files are stored outside of the HMI project, do one of the following:

   - Add the recipe file (*.rpp) that you saved at run time into the application using Add Component Into Application (for details, see page 18-3).
   - If the recipe file already exists in the application, you can just copy the modified file back into the application's RecipePlus folder. (For the path to the RecipePlus folder, see page 29-5.)

2. If recipe files are stored with the HMI project, and you have not already done so, convert the runtime application file to a development application, as described on page 14-6.

3. In the Explorer window in FactoryTalk View Studio, double-click the modified recipe file.

   The RecipePlus Editor opens.

4. Click the Ingredients tab.

5. If the data set you want to view is not visible, scroll right to see more data sets.

30 Using macros

This chapter describes:

- using macros to assign values to tags.
- using the Macros editor.
- when to use macros.
- running macros when tags or expressions change value.
- where to assign macros.

Using macros to assign values to tags

A macro is a list of tag assignments stored in a text file, in the format <tag>=<value>. Each assignment assigns a value to a tag. The value can be in the form of another tag, an expression, a numeric constant, or a string.

Examples: Using macros to set tag values

Tag1 = 8
Sets the value of Tag1 to 8.

Tank1\Message = “Tank1 overflow”
Sets the string tag Tank1\Message to Tank1 overflow.

Tag1 = Tag2
Sets the value of Tag1 to be the same as Tag2.

Tag1 = Tag1 + 1
Increases the value of Tag1 by 1.

Tag1 = if (Tag2 < Tag1) then 4 else 3
Performs the if-then-else calculation and stores the result in Tag1.

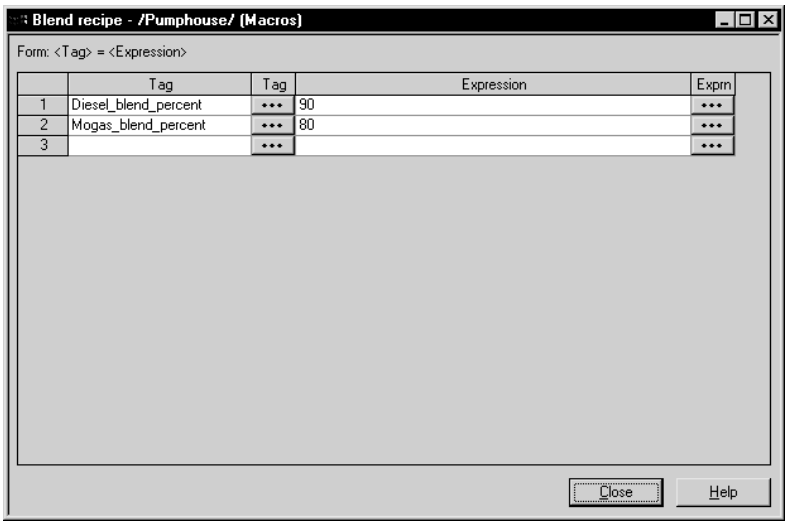
1Pump = {Industry-2} + {2Pump}
Adds the values of Industry-2 and 2Pump and stores the result in 1Pump.

Brackets surround Industry-2 because of the dash in the name. Brackets surround 2Pump because the name starts with a number. No brackets are used for 1Pump because this name is on the left side of the equal sign.

For more information about expression syntax, see Chapter 23.

Using the Macros editor

Use the Macros editor to create macros.



For details about using the Macros editor, see Help.

When to use macros

You can assign macros to run when:

- the application starts or shuts down.
- a graphic display opens or closes.
- a user logs in or out. Macros can be assigned to individual users and to groups of users.
- a specified tag or expression changes to a new non-zero value (using global connections).
- an operator presses a macro button.

At run time, when the macro runs, the values are sent to the tags at the data source.

At run time, the tag assignments are executed asynchronously. That is, the system does not wait for the completion of one tag assignment before executing the next. Therefore, do not rely on the order of assignments to control your process.

Running macros when tags or expressions change value

You can use global connections to run macros when tags or expressions change value. This means you can use the data source to trigger the macro to run.

FactoryTalk® View allows you to create up to five macros for use with global connections. The macros must be named Macro1, Macro2, Macro3, Macro4, and Macro5 in order to work with global connections.

For more information about global connections, see Chapter 8.

Example: Using macros to reset tag values

This example shows you how to run a macro whenever the operator needs to reset production information tags to known values.

The macro writes the desired values to the tags whenever the operator presses a momentary push button.

1. Create a memory tag called ResetProdData.
2. Create a momentary push button with the caption “Reset Production Data.” Assign the ResetProdData tag to the Value connection.
3. Create a macro called Macro1, with these tag assignments:

   TotalProductionUnits=0
   LineDownTime=0
   TotalRejects=0

4. In the Global Connections editor, assign the ResetProdData tag to the Remote Macro1 connection.

When the operator presses the Reset Production Data button, the value of the ResetProdData tag changes from 0 to 1. This tells FactoryTalk View to run Macro1, which writes the specified values to the tags in the macro.

Where to assign macros

Once you've created the macros you want to use, assign the macros in these editors:

In this editor	Do this
Startup	Assign application startup and shutdown macros.
Graphics	Assign macros to run when displays open or close, using the Display Settings dialog box.
Runtime Security	Assign macros to run when users log in and log out. Macros that you assign to groups of users run each time any member of the group logs in or logs out.
Global Connections	Specify the tags or expressions that will run the macros named Macro1 to Macro5.

APPENDIX A

Converting PanelBuilder 1400e applications

This appendix describes:

- terms that are different in PanelBuilder™ 1400e and FactoryTalk® View.
- steps for converting PanelBuilder 1400e applications.
- names of equivalent graphic objects in the two products.
- PanelBuilder 1400e graphic objects that are not supported in FactoryTalk View.
- PanelBuilder 1400e settings and controls that are not supported in FactoryTalk View.
- how communications are converted and which PanelBuilder 1400e communication protocols are not supported in FactoryTalk View.
- converting PanelBuilder 1400e Remote I/O communications.
- PanelBuilder 1400e graphic object features that are not supported in FactoryTalk View, with information about how to achieve the same result when possible.
- converting PanelBuilder 1400e expressions.

PanelBuilder 1400e applications are applications you create using PanelBuilder 1400e Configuration Software for Windows®. For information about converting applications from PanelBuilder or PanelBuilder32, see Appendix B.

You can convert PanelView 1200 applications to PanelBuilder 1400e applications, and then convert the PanelBuilder 1400e applications to FactoryTalk View Machine Edition applications.

Terminology

This section describes terms that are different in PanelBuilder 1400e and FactoryTalk View.

PanelBuilder 1400e term	FactoryTalk View term
screen	display, graphic display
Optional Keypad Write Expression	Optional Expression
programmable controller	data source

PanelBuilder 1400e term	FactoryTalk View term
control	connection

In FactoryTalk View, the data source can be memory or a device such as a programmable controller or an OPC® server. FactoryTalk View writes values to and reads values from the data source. The data source is configured to exchange information (in the form of numeric or string values) between FactoryTalk View and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Summary of steps

Follow these steps to convert PanelBuilder 1400e applications:

1. Prepare the application in PanelBuilder 1400e, and then convert the application file, as described in the next section.
2. Specify additional project settings, as described on page 4-11.

   For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the Project Settings editor.

   We recommend that you use the Project Settings editor to change the project window size, rather than using the Convert to new window size option in the Machine Edition Import Wizard.
3. If you use the Convert to new window size option in the Machine Edition Import Wizard, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.

   For more information, see page A-9.
5. Set up graphic object features that don't convert directly.

   For more information, see page A-12.
6. Check each expression you used in PanelBuilder 1400e.

   For more information, see page A-13.
7. If you are going to use a printer at run time, set it up for Ethernet® or USB printing. Adjust the printer settings on the PanelView™ Plus or PanelView™ Plus CE terminal.

   For information about setting up printers on the terminal, see the *PanelView Plus Terminals User Manual*. This manual is available on the FactoryTalk View Machine Edition CD.

Converting PanelBuilder 1400e application files

Follow these steps to convert a PanelBuilder 1400e application file, with the extension .pvc, to an FactoryTalk View application file, with the extension .med. The original PanelBuilder 1400e application file is not modified by the conversion.

For information about converting RIO applications, see page A-10.

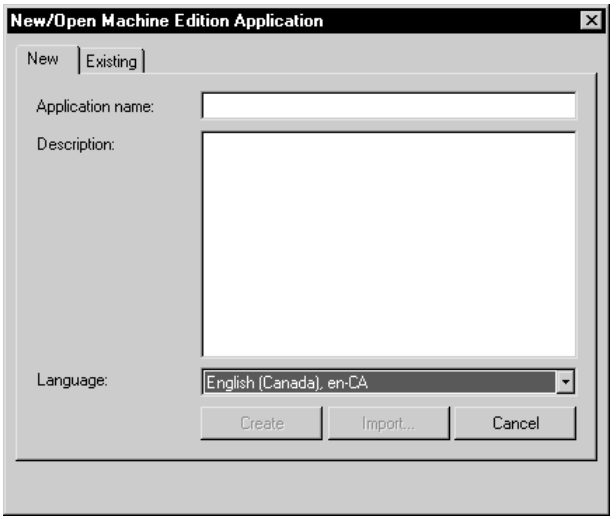
Steps to take in PanelBuilder 1400e before you convert the application

1. Delete the Pass-Through file assignment. FactoryTalk View Studio does not support pass-through file transfers.
2. Make sure the block transfer file numbers are sequential without gaps. If necessary, renumber the block transfer file numbers so there are no missing numbers. Tag addresses in the application will change automatically to match the new number.
3. Save the application.

You can convert the PanelBuilder 1400e application when you open FactoryTalk View Studio, or once FactoryTalk View Studio is already open.

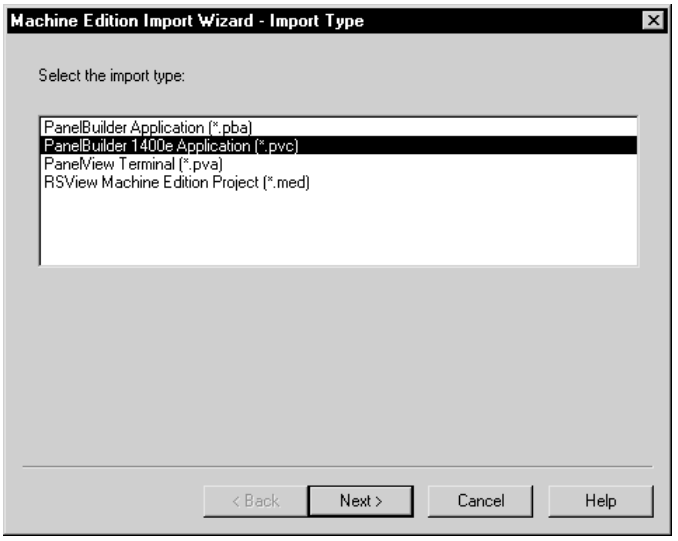
To convert a PanelBuilder 1400e application when you open FactoryTalk View Studio

1. Open FactoryTalk View Studio.



2. In the New tab, in the Application name box, type a name for your converted application, up to 32 characters long.
3. If desired, type a description of the application.

   If the PanelBuilder 1400e application contains an Application File Comment, the Application File Comment will overwrite the description you type here. You can add or change the description later, as described on page 4-17.
4. Specify a language for the converted application. For information about using different languages, see Chapter 12.
5. Click Import.



6. Follow the steps in the Machine Edition Import Wizard.

   For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, FactoryTalk View Studio converts the PanelBuilder 1400e application, creates the converted application's folders and files, and then displays the converted application in the Explorer window in FactoryTalk View Studio.

If there are any messages about conversion, they are displayed automatically in the Project Status dialog box.

The converted application is created in the ME\HMI projects directory, in a folder with the same name as the application name you specified in step 2.

This is the path to the ME\HMI projects directory:

\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects (Windows 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME\HMI projects (Windows XP or Windows Server 2003 R2)

Conversion messages are saved in a file called Convert.log, in the HMI projects directory.

To convert a PanelBuilder 1400e application when FactoryTalk View Studio is already open



New Application

1. On the File menu, click New Application, or click the New Application tool.

   If an application is already open, FactoryTalk View Studio asks you whether to close the application that is currently open. Click Yes.

2. Follow steps 2 through 6 in the previous procedure.

Equivalent graphic objects

This section describes graphic objects that are equivalent in PanelBuilder 1400e and FactoryTalk View, but have different names in the two products.

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
Increment Value Button	Ramp button	During conversion the button is set up to increment.
Decrement Value Button	Ramp button	During conversion the button is set up to decrement.
Increment Value Button with Display	Ramp button and numeric display	The Increment Value Button with Display is divided into two separate FactoryTalk View objects.
Decrement Value Button with Display	Ramp button and numeric display	The Decrement Value Button with Display is divided into two separate FactoryTalk View objects.
ASCII Input (small and large)	String input enable button	
Numeric Entry Keypad (small and large)	Numeric input enable button	
Screen List Selector’s list	Display list selector	The PanelBuilder 1400e Screen List Selector is divided into four separate FactoryTalk View graphic objects.
Screen List Selector’s Enter Key	Enter button	

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
Screen List Selector's Down Cursor	Move down button	
Screen List Selector's Up Cursor	Move up button	
Control List Selector's list	Control list selector	The PanelBuilder 1400e Control List Selector is divided into four separate FactoryTalk View graphic objects.
Control List Selector's Enter Key	Enter button	
Control List Selector's Down Cursor	Move down button	
Control List Selector's Up Cursor	Move up button	
Screen Select Keypad (small and large)	Display list selector	Specify the graphic displays that the display list selector can open.
Screen Keypad Enable Button	Display list selector	Specify the graphic displays that the display list selector can open.
Goto Screen Button	Goto display button	
Return to Previous Screen Button	Return to display button	
ASCII Display	String display	
Numeric Keypad Enable Button	Numeric input enable button	
Normally Open Momentary Push Button	Momentary push button	During conversion the button is set up to be normally open.
Normally Closed Momentary Push Button	Momentary push button	During conversion the button is set up to be normally closed.
Screen Print Button	Display print button	
Alarm History Sort By Time/Sort By Value Button	Sort alarms button	
Alarm Status Reset Qty/Time Button	Reset alarm status button	
Alarm Panel	Alarm banner	
Single Line Alarm Window	Alarm banner	

This PanelBuilder 1400e object	Is converted to this FactoryTalk View object	Notes
Alarm Status Screen	Alarm status list	
Clear All Button	Clear alarm history button	
Print Button (in Alarm History screen)	Print alarm history button	
Print Button (in Alarm Status screen)	Print alarm status button	
Alarm Status Button/Alarm History Button	Goto display button	
Exit Button	Close display button	
Alarm History List	Alarm list	
Display Mode Button	Alarm status mode button	
Time Display	Time and date display	During conversion the display is set up to show the time only. The PanelBuilder 1400e time format is not converted. For details about the FactoryTalk View time format, see page 21-16.
Date Display	Time and date display	During conversion the display is set up to show the date only. The PanelBuilder 1400e date format is not converted. For details about the FactoryTalk View date format, see page 21-16.
Arc (with solid fill style)	Arc (with solid back style) and line	The line graphic object is added because the solid FactoryTalk View arc shape does not have a line between the two points of the arc.



FactoryTalk View arc



FactoryTalk View arc with line

Unsupported graphic objects

These PanelBuilder 1400e objects are not supported in FactoryTalk View:

- Scrolling List (includes Cursor List, Multistate Indicator Object List, Local Message Object List, Numeric Data Display Object List)
- Set Bit Cursor Point

Unsupported settings and controls

This section describes PanelBuilder 1400e settings and controls that are not used in FactoryTalk View.

Controls for transferring runtime application files

PanelBuilder 1400e uses these optional controls for transferring files to the runtime terminal:

- Transfer Inhibit control
- Transfer Request control
- Transfer Status control

These controls are not necessary in FactoryTalk View because the ME Transfer Utility allows you to transfer the runtime project file while running a project on the runtime terminal.

Settings and controls for alarms

FactoryTalk View does not use these PanelBuilder 1400e features and settings to manage alarms:

- alarm relays
- bit alarm acknowledgement
- Remote Alarm Operation Hold Time. The PanelBuilder 1400e Remote Alarm Ack Control Hold Time will be used for all alarm hold times. You can change the hold time in the FactoryTalk View Alarm Setup editor, in the Advanced tab.
- Remote Alarm Control Delay Time. In FactoryTalk View, if an Ack connection is assigned, when an alarm is acknowledged the Ack connection is set immediately, without waiting for a delay time.

FactoryTalk View does not use these PanelBuilder 1400e controls to manage alarms:

- PLC Controlled Relay control
- PLC Controlled Audio control
- Acknowledge to PLC control (if the Alarm Acknowledge to PLC option is set to Bit)

Invalid characters in screen names

Characters in PanelBuilder 1400e screen names that are not supported in FactoryTalk View are replaced with the underscore character.

Screen security settings

PanelBuilder 1400e screen security settings are not converted, because FactoryTalk View uses a different method to assign security to graphic displays. For information about setting up security in FactoryTalk View, see Chapter 11.

Block tags

Block tags are not supported in FactoryTalk View. Block tags that are monitored for alarms in your PanelBuilder 1400e application are converted to bit arrays. For information about monitoring bit arrays for alarm conditions, see Chapter 9.

Converting non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted. For information about converting RIO communications, see page A-10.

FactoryTalk View does not use nodes for communications. Nodes are converted to RSLinx® topics. Topics are then converted into device shortcuts, to run with RSLinx® Enterprise™. You must have both RSLinx® Classic™ and RSLinx Enterprise installed to make this two-step conversion.

Tags are converted to HMI device tags and RSLinx aliases. The Unsolicited_Msgs node is not converted.

If you import an application multiple times, delete the device shortcuts in RSLinx Enterprise before re-importing. Otherwise, multiple unused device shortcuts will be created in RSLinx Enterprise.

For more information about setting up communications, see Chapter 5.

Unsupported tag data types

These tag data types are not supported in FactoryTalk View:

- Bit Position
- 1-BCD, 2-BCD, 5-BCD, 6-BCD, 7-BCD, 8-BCD
- BIN3, BIN4, BIN6, BIN8 (used with Modbus communications)

Tags that use these data types are converted to analog HMI tags with the Default data type. The Default data type uses floating point values.

For Bit Position data types, use the bitwise expression operators to display data that does not reference supported lengths. For more information, see the Rockwell Automation KnowledgeBase.

To open the KnowledgeBase

1. In FactoryTalk View Studio, on the Help menu, select Rockwell Software on the Web, and then click Rockwell Automation KnowledgeBase.

For information about using bitwise expression operators, see page 23-9.

Unsupported initial values

Device tags in FactoryTalk View do not use initial values. Memory tags are converted with their initial values.

Converting RIO communications

You can use Remote I/O (RIO) communications on the PanelView Plus and PanelView Plus CE runtime platforms.



RIO communications are not supported for applications that will run on a personal computer. However, you can test run your RIO applications on the development computer.

To convert an RIO application from PanelBuilder 1400e to FactoryTalk View

1. Convert the application, as described on page A-3.
2. Open the RSLinx Enterprise data server, and then double-click Communication Setup.
3. In the Communication Setup editor, add an RIO driver.
   - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
   - For all other PanelView Plus or PanelView Plus CE terminals, use the 2711P-RN6 driver.

   For information about adding drivers in RSLinx, see RSLinx Help.
4. Expand the RIO tree, right-click RIO Data, and then click Configure RIO.
5. In the RIO Configuration dialog box, right-click RIO, and then click Import.
6. Browse to the location of the RIO configuration file.

   The file is saved in the root of the application's directory.
7. In the Communications Setup editor, create a device shortcut that points to the RIO data device.

For information about creating a device shortcut, see RSLinx Help.

8. Apply the shortcut to the RIO driver.
9. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red “x” icons.

If red “x” icons appear after you import the RIO .xml file, you can fix block tags in the Communication Setup editor in FactoryTalk View. The length of block tags must be the same for Read and Write pairs sharing the same rack, group, and slot.

10. Create an alias for any data that is not a 16-bit integer or bit.
11. Save the converted RIO application.

RIO configurations are not saved with the application when you exit FactoryTalk View Studio. However, they are backed up with the application in the Application Manager. For information about handling multiple applications with different RIO settings, see the Rockwell Automation KnowledgeBase. For information about using the Application Manager, see page 4-10.

Unsupported PanelBuilder 1400e RIO tags

A PanelBuilder 1400e RIO tag will be converted to an HMI memory tag and an error will be logged to the conversion log file if the RIO tag:

- has a blank address.
- has a data type of 1-BCD, 2-BCD, 3-BCD, 5-BCD, 6-BCD, or 7-BCD.
- has a data type of Bit Position and its address does not reference a single bit.
- float has a data type of Float and its address has a bit offset assigned that is not 0.
- has a data type of Long Integer or 8 Digit BCD, its address has a bit offset assigned that is not 0, and its length or range is greater than (32 - *Start Bit*).
- has a tag type of Block.
- has an invalid PanelBuilder 1400e address or references an undefined rack.

Unsupported graphic object features

This section describes features of PanelBuilder 1400e graphic objects that are not supported in FactoryTalk View. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Image, text, arc, ellipse, line, panel, rectangle, wedge	Blinking wallpaper objects	If you want an object to blink at run time, unlock the wallpaper. In FactoryTalk View, all of the listed objects except images and panels use color animation to blink. For details, see page 22-9. Panels use the Blink property to blink. Color images do not blink. Monochrome images use the Blink property to blink.
Numeric Display	Polarity	If a PanelBuilder 1400e application was set up with the Polarity control requiring a negative number to display the minus sign, the numeric display will not work properly after the application is converted to FactoryTalk View Machine Edition.
Numeric Input Cursor Point, Numeric Data Display	Fixed Position and PLC Controlled decimal display options	Use an expression to achieve the same result. Assign the expression to the object's Value connection. For information about expressions, see Chapter 23.
Numeric Input Cursor Point, Numeric Keypad Enable Button, Numeric Keypad	PLC Controlled and Decimal Key Controlled input options	Objects are converted with the Decimal Point property set to Implicit.
Numeric Input Cursor Point	Retain Cursor on Cancel	The numeric input cursor point retains focus when the operator cancels entering a numeric value.
Maintained Push Button, Multistate Push Button, Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' connections. For information about macros, see Chapter 30. Assign the macro in the Startup editor. For details, see Help.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Trend	Blinking pens	
	Date labels on the X-Axis	The date is displayed in the title.
	Background screen plotting	You can plot tag values in the background by assigning the tags to a data log model. Tags set up for background screen plotting are automatically assigned to a data log model on conversion. However, data log models do not plot expression values. Therefore, expressions set up for background screen plotting are not converted. For information about data logging, see Chapter 26.
All objects	PanelBuilder 1400e object name	Object names are replaced with the FactoryTalk View default object names. The PanelBuilder 1400e object name is used for the object's description. You can view and edit the name and description in the Property Panel. For details, see Help.
All objects	Caption and image placement	FactoryTalk View supports one, three, or nine positions for captions and images, depending on the type of object. On conversion, captions and images are positioned using the closest match. Therefore some captions might overlap images, some captions might be truncated, and some images might be clipped to fit the object.
All objects	Multiple image labels	FactoryTalk View supports one image label per object or state. If a PanelBuilder 1400e object is set up to use multiple image labels, only the top left image is converted.

Converting expressions

Some PanelBuilder 1400e expression syntax is not supported in FactoryTalk View. Expressions are converted without modification, and then turned off by placing warning text at the beginning of the first line of the expression. In addition, exclamation marks (!) are placed at the beginning of each subsequent line of the expression. Warming text is also placed in expressions assigned to alarm triggers in the Alarm Setup editor. To turn on the expression, you must remove the warning text and exclamation marks, and revise the syntax if necessary.

The maximum expression length in FactoryTalk View is 16,000 characters. If a PanelBuilder 1400e expression contains more than 16,000 characters, the excess characters are not converted.

Some PanelBuilder 1400e objects support both tags and expressions. For these objects, if the text assigned to a connection could be valid syntax for both a tag and an expression, the connection is treated as an expression, and is therefore turned off.

For example, N20-0_String_64 could be the name of a tag, or it could be an expression that subtracts "0_String_64" from the tag "N20." The text would be converted as an expression, and turned off.

To turn on an expression

1. Select the object containing the expression.
2. Open the Property Panel, and then click the Connections tab.
3. In the Exprn column, click the Browse button beside the expression to turn on.



Browse button

4. In the Expression editor, delete the warning text and exclamation marks.
5. Revise the expression, if necessary, using the tables in the following three sections as guides.
6. Click Check Syntax.

For more information about using the Expression editor, see Chapter 23 or Help.

Equivalent expression syntax

This table describes FactoryTalk View expression syntax that is equivalent to PanelBuilder 1400e syntax. When you edit the converted expressions, replace the PanelBuilder 1400e syntax with the FactoryTalk View equivalent.

Syntax that is not listed in this table or in the next section is okay the way it is.

Type of expression component or operator	PanelBuilder 1400e syntax	FactoryTalk View syntax
Comment	REM or ‘	!
Line continuation	_ (underscore)	Not needed.
Equality	(=)	EQ or ==

Type of expression component or operator	PanelBuilder 1400e syntax	FactoryTalk View syntax
Bitwise Not	Not	~ (tilde)
Bitwise And	And	&
Bitwise Or	Or	| (pipe)
Bitwise XOr	XOr	^
If both operands are Byte, Integer, Long, Variant, or any combination of these data types, use the FactoryTalk View syntax. For other data types, no change is needed.		

Unsupported expression syntax

This table describes the PanelBuilder 1400e expression syntax that is not supported in FactoryTalk View with information about how to achieve the same result where possible.

Type of expression component or operator	PanelBuilder 1400e syntax	Equivalent FactoryTalk View syntax (if any)
Exit statement	Exit	
Local variables	DIM *varname* AS ... *varname* =	
Integer division	\	(*x* - (*x* MOD *y*))/*y*
Endif	If then endif If then else endif	If then else 0 If then else
Select case	Select Case Case1...CaseN CaseElse EndSelect	Use nested if-then-else.
Logical Xor (if one or both operands are Boolean or Single data types)	Xor	NOT ((*x* AND *y*) OR NOT (*x* OR *y*))

Order of precedence

The order of precedence is slightly different in FactoryTalk View. Check your expressions to make sure the result is what you intend.

PanelBuilder 1400e order of precedence	FactoryTalk View order of precedence
( )	( )
- (negation)	NOT, ~ (tilde)
*, / (floating point division)	*, /, MOD, %, **, AND, &&, &, >>, <<
\ (integer division)	+, -, OR, ||, |, ^
MOD	EQ, ==, NE, <>, LT, <, GT, >, LE, <=, GE, >=
+, - (subtraction)	
=, <>, <, >, <=, >=	
Not	
And	
Or	
Xor	

For more information about order of precedence, see page 23-11.

APPENDIX B

Converting PanelBuilder and PanelBuilder32 applications

This appendix describes:

- terms that are different in PanelBuilder™ and FactoryTalk® View.
- steps for converting PanelBuilder applications.
- names of equivalent graphic objects in PanelBuilder and FactoryTalk View.
- PanelBuilder graphic objects that are not supported in FactoryTalk View.
- PanelBuilder settings and controls that are not supported in FactoryTalk View.
- how communications are converted and which PanelBuilder communication protocols are not supported in FactoryTalk View.
- converting PanelBuilder Remote I/O communications.
- PanelBuilder graphic object features that are not supported in FactoryTalk View, with information about how to achieve the same result when possible.

This appendix uses the term PanelBuilder to refer to both PanelBuilder and PanelBuilder32 features.

For information about converting applications from PanelBuilder 1400e, see Appendix A.

Terminology

This section describes terms that are different in PanelBuilder and FactoryTalk View.

PanelBuilder term	FactoryTalk View term
screen	display, graphic display
programmable controller	data source
control	connection

In FactoryTalk View, the data source can be memory or a device such as a programmable controller or an OPC® server. FactoryTalk View writes values to and reads values from the data source. The data source is configured to exchange information (in the form of

numeric or string values) between FactoryTalk View and the machine that your application is controlling. The general term data source is used unless specifically discussing a programmable controller.

Summary of steps

Follow these steps to convert PanelBuilder applications:

1. Convert the application file, as described in the next section.
2. Specify additional project settings, as described on page 4-11.

   For example, if you want the application to have a border around its graphic displays, or to use a title bar, you can specify these options in the Project Settings editor.
3. If you select Convert to new window size, check the position of the graphic objects in each display.
4. Set up communications and edit tags that don't convert directly.

   For more information, see page B-7.
5. Set up graphic object features that don't convert directly.

   For more information, see page B-10.

Converting PanelBuilder application files

Follow these steps to convert a PanelBuilder application file, with the extension .pba or .pva, to an FactoryTalk View application file, with the extension .med. The original PanelBuilder application file is not modified by the conversion.

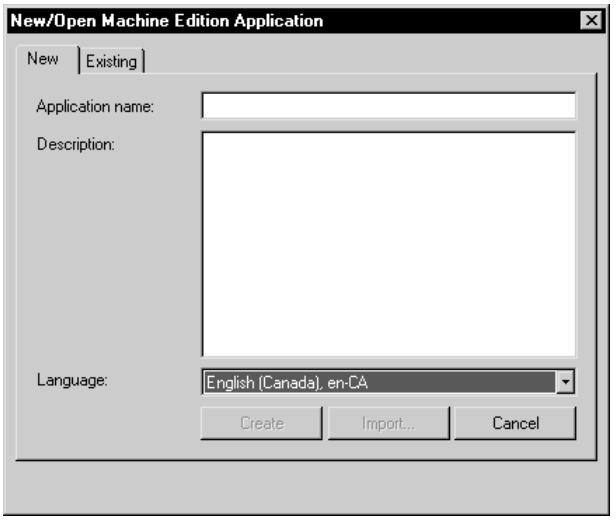
Steps to take in PanelBuilder before you convert the application

1. Semicolons (;) in tag addresses are supported in PanelBuilder, but not in FactoryTalk View Studio. Before importing the PanelBuilder application, in the PanelBuilder Tag Editor, change the semicolons to colons (:).
2. Dashes (-) in tag names are supported in PanelBuilder, but not in FactoryTalk View Studio. Before importing the PanelBuilder application, in the PanelBuilder Tag Editor, locate any tags whose names contain dashes and duplicate the tags. Then rename the tags without the dash, or replace the dash with an underscore (_). Once the tags have been renamed, use the Tag Search feature to find the graphic objects using the original tag names and edit the objects to replace the old tag names with the new ones.

You can convert the PanelBuilder application when you open FactoryTalk View Studio, or once FactoryTalk View Studio is already open.

To convert a PanelBuilder application when you open FactoryTalk View Studio

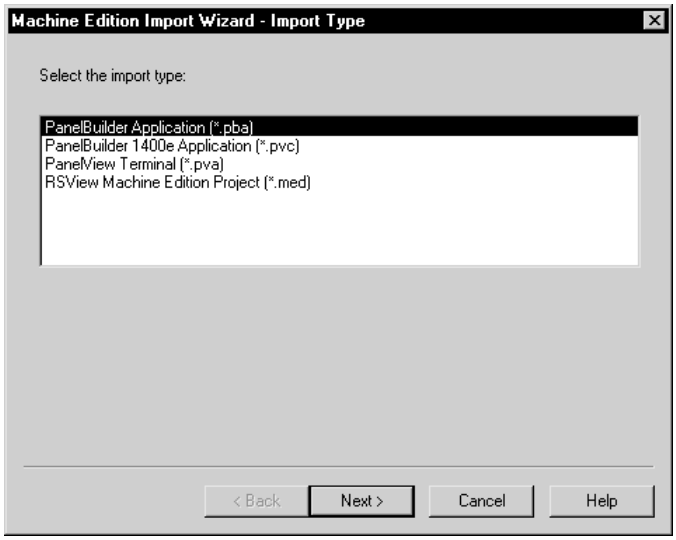
1. Open FactoryTalk View Studio.



2. In the New tab, in the Application name box, type a name for your converted application, up to 32 characters long.
3. If desired, type a description of the application.

   If the PanelBuilder application contains an Application Description, the Application Description will overwrite the description you type here. You can add or change the description later, as described on page 4-17.
4. Specify the last language that was used to edit the application. This will be used for the converted application.

   You can only import one language for your application, even if the original application uses multiple languages. The imported language will be the last language used to edit the application. For information about using different languages, see Chapter 12.
5. Click Import.



6. Follow the steps in the Machine Edition Import Wizard.

   For details about the options in the Machine Edition Import Wizard, see Help.

When you complete the steps of the wizard, FactoryTalk View Studio converts the PanelBuilder application, creates the converted application's folders and files, and then displays the converted application in the Explorer window in FactoryTalk View Studio.

If there are any messages about conversion, they are displayed automatically in the Project Status dialog box.

The converted application is created in the ME\HMI projects directory, in a folder with the same name as the application name you specified in step 2.

This is the path to the ME\HMI projects directory:

\Documents and Settings\All Users\Documents\RSView Enterprise\ME\HMI projects (Windows® 2000)

or

\Documents and Settings\All Users\Shared Documents\RSView Enterprise\ME\HMI projects (Windows XP or Windows Server 2003 R2)

Conversion messages are saved in a file called Convert.log, in the HMI projects directory.

To convert a PanelBuilder application when FactoryTalk View Studio is already open



New Application

1. On the File menu, click New Application, or click the New Application tool.

   If an application is already open, FactoryTalk View Studio asks you whether to close the application that is currently open. Click Yes.

2. Follow steps 2 through 6 in the previous procedure.

Equivalent graphic objects

This section describes graphic objects that are equivalent in PanelBuilder and FactoryTalk View, but have different names in the two products.

This PanelBuilder object	Is converted to this FactoryTalk View object	Notes
Numeric Entry Keypad Enable Button	Numeric input enable button	
Numeric Entry Cursor Point	Numeric input enable button	
Increment/Decrement Entry Button	Numeric input enable button	The numeric input enable button is set up to work as a ramp button, using the Fine Step value. The Coarse Step value is not converted.
ASCII Entry Keypad Enable button	String input enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
ASCII Entry Cursor Point	String input enable button	The Show Current String on ASCII Scratchpad setting is not converted. The pop-up scratchpad or keyboard is always blank when opened.
Message Display	Multistate indicator	
Numeric Data Display	Text	The text object contains a numeric embedded variable that displays the read tag. If the original numeric data display object does not contain text, manually create a numeric display object with the desired properties in FactoryTalk View.
Connected Line	Polyline	

This PanelBuilder object	Is converted to this FactoryTalk View object	Notes
Circle	Ellipse	The ellipse has a circular shape.
Freeform	Freehand	
Screen List Selector	Display list selector	
Goto Screen Button	Goto display button	
Return Screen Button	Return to display button	
New Password Button	Password button	
Print Alarm List Button	Print alarm history button	
Clear Alarm List Button	Clear alarm history button	

Unsupported graphic objects

These PanelBuilder objects are not supported in FactoryTalk View:

- Print Only Object
- Circular Scale
- Scrolling Text
- Print Alarm Button
- Horn Silence Button
- Lamp/Horn Test Button
- Select Operator Button
- Enable/Disable Security Button
- Verify Password Button. The FactoryTalk View Password button opens a dialog box that allows the user to type and verify a new password.

Unsupported settings and controls

This section describes PanelBuilder settings and controls that are not used in FactoryTalk View.

Settings and controls for alarms

FactoryTalk View does not use these PanelBuilder features and settings to manage alarms:

- Ack setting for alarm messages; in FactoryTalk View, all alarms can be acknowledged
- bit alarm acknowledgement

FactoryTalk View does not use these PanelBuilder controls to manage alarms:

- Remote Ack All Handshake Tag
- Remote Clear All Alarm Tag
- Remote Clear All Alarm Handshake Tag

Invalid characters in screen names and tag names

Characters in PanelBuilder screen names and tag names that are not supported in FactoryTalk View are replaced with the underscore character.

Time and date

PanelBuilder time and date formats are not converted. For details about FactoryTalk View time and date formats, see page 21-16.

External fonts

PanelBuilder external fonts are not converted. When you convert your application you can specify the font to use instead. For details, see Help for the Machine Edition Import Wizard.

Screen security settings

PanelBuilder screen security settings are not converted, because FactoryTalk View uses a different method to assign security to graphic displays. For information about setting up security in FactoryTalk View, see Chapter 11.

Power-up options

These PanelBuilder power-up options are not imported into FactoryTalk View:

- Write Last Terminal State to Controller
- Display Last User Screen
- Use Terminal Presets

Converting non-RIO communications

This section describes how communications that do not use Remote I/O (RIO) are converted. For information about converting RIO communications, see page B-8.

FactoryTalk View does not use nodes for communications. Nodes are converted to RSLinx® topics. Topics are then converted into device shortcuts, to run with RSLinx® Enterprise™. You must have both RSLinx® Classic™ and RSLinx Enterprise installed to make this two-step conversion.

If you import an application multiple times, delete the device shortcuts in RSLinx Enterprise before re-importing. Otherwise, multiple unused topics will be created in RSLinx Enterprise.

For more information about setting up communications, see Chapter 5.

Tags for unsupported communication protocols

Some communication protocols are not supported in FactoryTalk View. For example, DH+™ communications that use the AutoMax node type are not supported. FactoryTalk View does not support CIP and Assembly Object addressing (used in PanelBuilder32 Ethernet communications).

Tags that use unsupported communication protocols are converted to HMI memory tags. Once you have set up communications for your converted application, change the memory tags to device tags that point to the correct addresses. All other imported tags are converted to HMI device tags.

For information about editing HMI tags, see Chapter 7. For more information about which communication protocols are not supported, see Help or see the Rockwell Automation Knowledgebase.

Bit array tags

You can monitor bit arrays for alarm conditions in FactoryTalk View, but you can't assign bit arrays to most graphic objects or write to bit arrays. (The only exception is the piloted control list selector object. For this object, you can assign a bit array tag to the Visible States connection.)

All bit array tags in your PanelBuilder application are converted to HMI memory tags.

For information about monitoring bit arrays for alarm conditions, see Chapter 9. For information about editing HMI tags, see Chapter 7. For information about the piloted control list selector, see Help.

Converting RIO communications

Remote I/O (RIO) communications are not supported for applications that will run on personal computers. You can use RIO communications on the PanelView™ Plus and PanelView™ Plus CE runtime platforms.

Supported PanelBuilder tags are converted to HMI device tags and RSLinx aliases. For information about unsupported PanelBuilder tags, see below.

To convert an RIO application from PanelBuilder to FactoryTalk View

1. Convert the application, as described on page B-2.
2. Open the RSLinx Enterprise data server, and then double-click Communication Setup.
3. In the Communication Setup editor, add an RIO driver.
   - For PanelView Plus 400 and 600 terminals, use the 2711P-RN1 driver.
   - For all other PanelView Plus or PanelView Plus CE terminals, use the 2711P-RN6 driver.

   For information about adding drivers in RSLinx, see RSLinx Help.
4. Expand the RIO tree, right-click RIO Data, and then click Configure RIO.
5. In the RIO Configuration dialog box, right-click RIO, and then click Import.
6. Browse to the location of the RIO configuration file.

   The file is saved in the root of the application's directory.
7. In the Communications Setup editor, create a device shortcut named "PVRIO" that points to the RIO data device.

   For information about creating a device shortcut, see RSLinx Help.
8. Correct any invalid RIO configurations. Invalid RIO configurations are highlighted with red "x" icons.
9. Save the converted RIO application.

Unsupported PanelBuilder RIO tags

A PanelBuilder RIO tag will be converted to an HMI memory tag and an error will be logged to the conversion log file if the RIO tag:

- has a blank address.
- has a data type of Bit Array and its address does not have an array size of 1, 8, 16, or 32.

- has a data type of Bit Array, and its array size is 16 or 32, but its address does not have a bit offset of 0.
- has a data type of Bit or BOOL, and its address does not contain the bit delimiter character "/".
- has a data type of 4-BCD, Unsigned Integer, Signed Integer or INT, Character Array, or DINT, and its address contains the bit delimiter character "/".
- is a block transfer tag with a data type of Bit Array, and its array size is 8, but its address does not have a bit offset of 0 or 8.
- is a block transfer tag with a data type of SINT, and its address does not have a bit offset of 0 or 8.
- is an I/O tag with an address that references an undefined rack.
- is an I/O tag with a data type of SINT, and its address does not have a bit offset of 0 or 10.
- does not have a valid I/O address or block transfer address.

Unsupported graphic object features

This section describes features of PanelBuilder graphic objects that are not supported in FactoryTalk View. The Notes column provides additional information and describes methods for achieving the same result when possible.

Graphic object	Unsupported feature in FactoryTalk View	Notes
Image text, arc, ellipse, freehand, line, polyline, rectangle, wedge	Blink property	In FactoryTalk View, all of the listed objects except images use color animation to blink. For details, see page 22-9. Color images do not blink. Monochrome images use the Blink property to blink.
Increment/Decrement Entry Button (Converted to numeric input enable button)	Allow Home/End Allow Wrap Ramping by coarse steps	

Graphic object	Unsupported feature in FactoryTalk View	Notes
Maintained Push Button, Multistate Push Button, Standard Control List Selector	Initial state values	If you want to set these objects' states on application startup, create a macro to set the appropriate tag values for the objects' connections. For information about macros, see Chapter 30. Assign the macro in the Startup editor. For details, see Help.
Multistate Indicator, Message Display	Print Setting	
Bar Graph	Inner text and inner graphic	Converted to a separate text object and image object.
Gauge	Inner text and inner graphic	Converted to a separate text object and image object.
	Scale clipping	If the scale doesn't fit within the height or width of the gauge, it is not clipped. Check the position of the scale to ensure it doesn't overlap other objects.
	Needle	Converted to a separate gauge object; if the gauge had 2 needles, each needle is converted to a separate gauge object.
Alarm List	No Acknowledgement Required	All alarms can be acknowledged.
All objects	Image placement	FactoryTalk View supports one, three or nine positions for images, depending on the type of object. On conversion, images are positioned using the closest match. Therefore some images might be clipped to fit the object.
All objects	Turn Object View On property	If this property is set to False, the converted object has a transparent background, no border, no caption, and no image.
All objects	Blinking inner graphics	If the inner graphic uses a color image, it will not blink. Use a monochrome image if you want the inner graphic to blink.

APPENDIX C

System tags

This appendix describes system tags.

System tags are preconfigured HMI tags created by FactoryTalk® View. System tags are read-only. Display them as needed in your application.

Alarms

The following tag contains the time and date when the status of alarms was last reset. The date uses the long date format.

Tag name	Type	Function
system\AlarmReset DateAndTimeString	String	Contains the date and time of the last alarm reset.

For information about resetting alarms, see page 9-9.

Graphics

The following HMI tags can be used to make graphic objects appear as though they are blinking on and off:

Tag name	Type	Function
system\BlinkFast	Digital	Toggles on and off every 100 ms (10 times per second).
system\BlinkSlow	Digital	Toggles on and off every 500 ms (twice per second).

A more efficient way to make graphic objects blink is to use the blinking color option in color animation. For details, see page 22-9.

Also, many objects have a Blink property that you can set up. For information about specific objects, see Help.

Time

These HMI tags record time and date information in various formats:

Tag Name	Type	Provides this data	Read or write
system\Date	String	System date.	Read only
system\DateAndTime Integer	Analog	Number of seconds elapsed since midnight (00:00:00) January 1, 1970, coordinated universal time.	Read only
system\DateAndTime String	String	Complete date and time display. For example: Monday, December 12 2001 10:47:50 AM	Read only
system\DayOfMonth	Analog	Day of the month (1 - 31).	Read only
system\DayOfWeek	Analog	Day of the week (1-7); Sunday = 1.	Read only
system\DayOfYear	Analog	Day of the year (1-366).	Read only
system\Hour	Analog	Hour of the day (0-23).	Read and write
system\Minute	Analog	Minutes (0 - 59).	Read and write
system\Month	Analog	Number for month (1-12).	Read only
system\MonthString	String	Name of the month.	Read only
system\Second	Analog	Seconds (0 - 59).	Read and write
system\Time	String	System Time.	Read only
system\Year	Analog	The year (1980-2099).	Read only

For information about using the data source to update the system date and time, or about sending the runtime computer's date and time to the data source, see Chapter 8.

User

This tag contains the name of the current user:

Tag Name	Type	Function
system\User	String	Contains name of logged-in user.

We recommend that you use the expression security function CurrentUserName( ) instead of the system\User tag, especially if you intend to convert the application to FactoryTalk View Supervisory Edition. In distributed applications, system\User returns the name of the user logged into the HMI server, not the user logged into the display client.

For more information about the security functions, see page 23-15.

APPENDIX D

ODBC database schema

This appendix describes the ODBC database format, or schema, for messages from FactoryTalk® Diagnostics. The target table of the ODBC database to which you are sending messages must use the format shown in this appendix.

The option of logging FactoryTalk Diagnostics messages to an ODBC database is available for personal computers only.

For information about setting up FactoryTalk Diagnostics, see Chapter 10.

FactoryTalk Diagnostics log table

FactoryTalk Diagnostics log data in ODBC format uses one table.

This column	Contains	SQL data type	Length
TimeStmp	The time and date data was logged, in coordinated universal time format. Encoded as a date variant.	SQL_TIMESTAMP	Driver dependent
MessageText	Message to be logged.	SQL_VARCHAR, *or* SQL_CHAR	254
Audience	A number representing the message audience: 0 for Operator 1 for Engineer 2 for Developer 3 for Secure	SQL_SMALLINT, *or* SQL_INTEGER	1
Severity	A number representing the severity of the diagnostics message: 0 for Error 1 for Warning 2 for Information 3 for Audit	SQL_SMALLINT, *or* SQL_INTEGER	1
Area	The FactoryTalk path to the area in which the activity occurred. Used for FactoryTalk® View Site Edition only.	SQL_VARCHAR, *or* SQL_CHAR	80
Location	The name of the computer where the message was generated.	SQL_VARCHAR, *or* SQL_CHAR	15

This column	Contains	SQL data type	Length
UserID	The name of the user (including domain name, if there is one) that initiated the action that caused the diagnostics message. If the diagnostics message was caused by an HMI server, the user column contains "System."	SQL_VARCHAR, *or* SQL_CHAR	38
UserFullName	The full name of the user that was logged in when the activity occurred.	SQL_VARCHAR, *or* SQL_CHAR	255
Provider	The name of the product that generated the message.	SQL_VARCHAR, *or* SQL_CHAR	20

APPENDIX E

Importing and exporting alarm XML files

This appendix describes:

- creating alarm XML files.
- exporting, editing, and importing XML files.
- the alarm XML file structure.

About XML

XML is the Extensible Markup Language used to create documents with structured text information. It has a standardized format and structure. You can use XML to edit the elements and attributes needed to create an alarm setup file or to modify graphic displays. For information about working with graphics XML files, see Appendix H.

Sometimes editing your alarm setup in an XML file is quicker than working in FactoryTalk® View. For example, if you have a list of 100 tags to monitor for alarms, with multiple messages for each tag, you might prefer to enter all the information in a text editor, and then import the alarm setup information into FactoryTalk View.

Another example of using XML files is to export the alarm setup information you develop in one application, import the setup information to another FactoryTalk View application, and then modify the alarm setup as needed. Or, you could modify the information in the XML file before importing it.

For more information about XML, see the World Wide Web Consortium's web page about XML at:
http://www.w3.org/XML.

Creating alarm XML files by exporting

The quickest way to create an XML file for your application's alarm setup is to export the data from FactoryTalk View. You can then open the XML file in Notepad, make your changes, and import the file back into FactoryTalk View.

The strings for the application's current language are exported to the XML file. To export strings for another language, reopen the application in the new language and repeat the XML export.

To export alarm information to an XML file

1. In the Explorer window, right-click the Alarm Setup editor.
2. Click Import and Export.

The Alarm Import Export Wizard opens.

3. Follow the instructions in the wizard.

   For information about using the Alarm Import Export Wizard, see Help.

FactoryTalk View creates a file with the name you specify, in the location you specify.

Editing XML files

We recommend that you use Notepad to edit your XML files.

If you do not want to change a property, you don't need to include it in the XML file.

Saving XML files in Notepad

Save XML files created or edited in Notepad using either UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to UTF-16 file format. For files containing strings in English or other Latin-based languages, UTF-8 is recommended, to reduce the size of the XML file. For other languages such as Chinese, Japanese, or Korean, UTF-16 is recommended.

The first line of every XML file contains XML version and encoding attributes. Make sure the encoding attribute matches the format that you are going to use when you save the file. For example, if the original file was saved in UTF-8 format and you plan to save it in UTF-16 format, make sure the first line specifies encoding="UTF-16".

Testing XML files

An XML file must be well-formed to be imported. To find out whether your XML file is well-formed, test it.

To test an XML file

1. Open the XML file in Internet Explorer.

If you can see the XML code, your file is well-formed. If the XML code is not well-formed, Internet Explorer displays an error message.

Importing XML files

You can import an alarm setup that has been created using an external programming tool or editor, or you can import an XML file that you originally exported from FactoryTalk View and then modified.



When you import an alarm setup, your existing alarm setup will be overwritten. Back up your application first, using the Application Manager tool. Or, you can save a copy of your existing alarm setup by exporting it to an XML file before you import the new one.

Error log file

If errors occur during importing, the errors are logged to a text file. The file opens automatically when importing is finished. The last paragraph of the file lists the location of the file.

Importing alarm XML files

To import alarm information from an XML file

1. In the Explorer window, right-click the Alarm Setup editor.
2. Click Import and Export.

   The Alarm Import Export Wizard opens.
3. Follow the instructions in the wizard.

   For more information about using the Alarm Import Export Wizard, see Help.

Alarm setup XML file structure

The alarm setup XML file is a FactoryTalk View XML document that describes the alarm setup for an application. The root element of the XML document is called **alarms**. It represents the Alarm Setup editor. An XML document can contain only one root element. All other elements in the document must be contained or nested within the root element.

In an XML document, the start of an element is marked **<element name>**. The end is marked **</element name>**.

If the element contains no subelements, the end can be marked **/>**. For example, <trigger id="T1" type="value" ack-all-value="0" />.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes.

Here is a sample structure for an alarm XML document:.

Element	Description
<alarms>	Root element.
<alarm>	Contains attributes from the Advanced tab of the Alarm Setup editor, as well as the triggers and messages elements.
<triggers>	Contains a trigger element for each trigger in the Triggers tab of the Alarm Setup editor.
<trigger id="T1" />	Contains attributes for the first alarm trigger.

Element	Description
<trigger id="T2" />	Contains attributes for the second alarm trigger.
</triggers>	Indicates the end of the triggers element.
<messages>	Contains a message element for each message in the Messages tab of the Alarm Setup editor.
<message id="M1" />	Contains attributes for the first alarm message.
<message id="M2" />	Contains attributes for the second alarm message.
</messages>	Indicates the end of the messages element.
</alarm>	Indicates the end of the alarm element.
</alarms>	Indicates the end of the alarms element.

You can specify multiple attributes for an element. For example, the alarm element contains 11 possible attributes from the Advanced tab of the Alarm Setup editor.

For more information about alarm elements and their attributes, see Help for the Alarm Import Export Wizard.

APPENDIX F

RFC1766 names

This appendix describes RFC1766 names for Windows® languages.

Mapping languages to RFC1766 names

The following table lists the languages that Windows supports and the RFC1766 name associated with each language.

You can use the codes to name the translated application files before importing them. The codes are also used with the CurrentLanguage function.

RFC1766 Name	Language – Country/Region
af–ZA	Afrikaans – South Africa
sq–AL	Albanian – Albania
ar–DZ	Arabic – Algeria
ar–BH	Arabic – Bahrain
ar–EG	Arabic – Egypt
ar–IQ	Arabic – Iraq
ar–JO	Arabic – Jordan
ar–KW	Arabic – Kuwait
ar–LB	Arabic – Lebanon
ar–LY	Arabic – Lybia
ar–MA	Arabic – Morocco
ar–OM	Arabic – Oman
ar–QA	Arabic – Qatar
ar–SA	Arabic – Saudi Arabia
ar–SY	Arabic – Syria
ar–TN	Arabic – Tunisia
ar–AE	Arabic – United Arab Emirates
ar–YE	Arabic – Yemen
hy–AM	Armenian – Armenia
az–AZ–Cyrl	Azeri (Cyrillic) – Azerbaijan
az–AZ–Latn	Azeri (Latin) – Azerbaijan

RFC1766 Name	Language – Country/Region
eu–ES	Basque – Basque
be–BY	Belarusian – Belarus
bg–BG	Bulgarian – Bulgaria
ca–ES	Catalan – Catalan
zh–HK	Chinese – Hong Kong SAR (Default Sort Order – Stroke Count)
zh–HK	Chinese – Hong Kong SAR (Alternate Sort Order – Stroke Count)
zh–MO	Chinese – Macau SAR (Default Sort Order – Pronunciation)
zh–MO	Chinese – Macau SAR (Alternate Sort Order – Stroke Count)
zh–CN	Chinese – China (Default Sort Order – Pronunciation)
zh–CN	Chinese – China (Alternate Sort Order – Stroke Count)
zh–SG	Chinese – Singapore (Default Sort Order – Pronunciation)
zh–SG	Chinese – Singapore (Alternate Sort Order – Stroke Count)
zh–TW	Chinese – Taiwan (Default Sort Order – Stroke Count)
zh–TW	Chinese – Taiwan (Alternate Sort Order – Bopomofo)
hr–HR	Croatian – Croatia
cs–CZ	Czech – Czech Republic
da–DK	Danish – Denmark
div–MV	Dhivehi – Maldives
nl–BE	Dutch – Belgium
nl–NL	Dutch – The Netherlands
en–AU	English – Australia
en–BZ	English – Belize
en–CA	English – Canada
en–CB	English – Caribbean
en–IE	English – Ireland
en–JM	English – Jamaica
en–NZ	English – New Zealand
en–PH	English – Philippines
en–ZA	English – South Africa
en–TT	English – Trinidad and Tobago
en–GB	English – United Kingdom

RFC1766 Name	Language – Country/Region
en–US	English – United States
en–ZW	English – Zimbabwe
et–EE	Estonian – Estonia
fo–FO	Faroese – Faroe Islands
fa–IR	Farsi – Iran
fi–FI	Finnish – Finland
fr–BE	French – Belgium
fr–CA	French – Canada
fr–FR	French – France
fr–LU	French – Luxembourg
fr–MC	French – Monaco
fr–CH	French – Switzerland
mk–MK	FYRO Macedonian
gl–ES	Galician – Galician
ka–GE	Georgian – Georgia (Default Sort Order – Traditional)
ka–GE	Georgian – Georgia (Alternate Sort Order – Modern Sort)
de–AT	German – Austria
de–DE	German – Germany (Default Sort Order – Dictionary)
de–DE	German – Germany (Alternate Sort Order – Phone Book Sort DIN)
de–LI	German – Liechtenstein
de–LU	German – Luxembourg
de–CH	German – Switzerland
el–GR	Greek – Greece
gu–IN	Gujarati – India
he–IL	Hebrew – Israel
hi–IN	Hindi – India
hu–HU	Hungarian – Hungary (Default Sort Order)
hu–HU	Hungarian – Hungary (Alternate Sort Order – Technical Sort)
is–IS	Icelandic – Iceland
id–ID	Indonesian – Indonesia
it–IT	Italian – Italy

RFC1766 Name	Language – Country/Region
it–CH	Italian – Switzerland
ja–JP	Japanese – Japan (Default Sort Order)
ja–JP	Japanese – Japan (Alternate Sort Order – Unicode)
kn–IN	Kannada – India
kk–KZ	Kazakh – Kazakhstan
kok–IN	Konkani – India
ko–KR	Korean – Korea (Default Sort Order)
ko–KR	Korean – Korea (Alternate Sort Order – Korean Xwansung Unicode)
ky–KZ	Kyrgyz – Kazakhstan
lv–LV	Latvian – Latvia
lt–LT	Lithuanian – Lithuania
ms–BN	Malay – Brunei
ms–MY	Malay – Malaysia
mr–IN	Marathi – India
mn–MN	Mongolian – Mongolia
nb–NO	Norwegian (Bokml) – Norway
nn–NO	Norwegian (Nynorsk) – Norway
pl–PL	Polish – Poland
pt–BR	Portuguese – Brazil
pt–PT	Portuguese – Portugal
pa–IN	Punjabi – India
ro–RO	Romanian – Romania
ru–RU	Russian – Russia
sa–IN	Sanskrit – India
sr–SP–Cyrl	Serbian (Cyrillic) – Serbia
sr–SP–Latn	Serbian (Latin) – Serbia
sk–SK	Slovak – Slovakia
sl–SI	Slovenian – Slovenia
es–AR	Spanish – Argentina
es–BO	Spanish – Bolivia
es–CL	Spanish – Chile

RFC1766 Name	Language – Country/Region
es–CO	Spanish – Colombia
es–CR	Spanish – Costa Rica
es–DO	Spanish – Dominican Republic
es–EC	Spanish – Ecuador
es–SV	Spanish – El Salvador
es–GT	Spanish – Guatemala
es–HN	Spanish – Honduras
es–MX	Spanish – Mexico
es–NI	Spanish – Nicaragua
es–PA	Spanish – Panama
es–PY	Spanish – Paraguay
es–PE	Spanish – Peru
es–PR	Spanish – Puerto Rico
es–ES	Spanish – Spain (Default Sort Order – International)
es–ES	Spanish – Spain (Alternate Sort Order – Traditional)
es–UY	Spanish – Uruguay
es–VE	Spanish – Venezuela
sw–KE	Swahili – Kenya
sv–FI	Swedish – Finland
sv–SE	Swedish – Sweden
syr–SY	Syriac – Syria
ta–IN	Tamil – India
tt–RU	Tatar – Russia
te–IN	Telugu – India
th–TH	Thai – Thailand
tr–TR	Turkish – Turkey
uk–UA	Ukrainian – Ukraine
ur–PK	Urdu – Pakistan
uz–UZ–Cyrl	Uzbek (Cyrillic) – Uzbekistan
uz–UZ–Latn	Uzbek (Latin) – Uzbekistan
vi–VN	Vietnamese – Vietnam

APPENDIX G

Features supported in different versions of FactoryTalk View

This appendix describes:

- which versions of FactoryTalk® View ME Station are supported.
- which features are not supported in previous versions of FactoryTalk View ME Station.

Which versions are supported

FactoryTalk® View Studio allows you to create runtime (.mer) files for these versions of FactoryTalk View ME Station:

- FactoryTalk View ME Station version 5.00
- RSView® ME Station version 4.00
- RSView ME Station version 3.20
- RSView ME Station version 3.10
- RSView ME Station version 3.00

Multiple version support is useful for system designers and others who create and modify applications for different versions of FactoryTalk View ME Station on an ongoing basis. You can use the latest version of FactoryTalk View Studio on a single development computer to provide applications for terminals that use previous versions of FactoryTalk View ME Station.

To check which version of FactoryTalk View ME Station you are using:

1. In FactoryTalk View ME Station, click Terminal Settings.
2. Click System Information.
3. Click About FactoryTalk View ME Station.

Creating runtime application files for previous versions

When you create the runtime application file (with the file extension .mer), you can specify the version of FactoryTalk View ME Station for which to create the file. For

example, if the application will run on a terminal that uses RSView ME Station version 3.20, you can specify that version for the .mer file.

If the application contains features that are not supported by the version you select, FactoryTalk View displays a validation report that lists the unsupported features. The runtime application file is not created. You must remove or turn off the unsupported features before you can create the runtime application file.

For information about creating runtime application files, see Chapter 14.

The remainder of this appendix lists the features that are not supported in previous versions of FactoryTalk View ME Station. The tables also show how to remove or replace the unsupported features.

Features that are not supported in version 4.00 or earlier

These version 5.00 features are not supported in version 4.00 of RSView ME Station, nor in earlier versions. The right column describes how to remove or replace the feature.

To remove or replace this feature	Do this
Delete action assigned to a RecipePlus button graphic object	Delete the button or change the action to Download, Upload, Upload and Create, Restore, or Save.
Rename action assigned to a RecipePlus button graphic object	Delete the button or change the action to Download, Upload, Upload and Create, Restore, or Save.
"Display undefined strings using the default language" option	In the Language Configuration dialog box, clear the option's checkbox.
Tag assigned to the MERuntime RAM Usage, Total RAM Usage, or Available RAM connection	Unassign the tag in the Memory tab of the Global Connections editor.
Tag or expression assigned to the Remote RAM Usage connection	Unassign the tag or expression in the Memory tab of the Global Connections editor.
CaptionOnBorder property assigned to graphic object	In the Property Panel, set the object's CaptionOnBorder property to False.
Embedded variable "L" character assigned to a caption or message	Select the Tag radio button in the Numeric Variable or String Variable dialog box for the caption or message.
Minus sign (-) qualifier used to show right-most text assigned to a caption or message	Clear the "Right-most characters in string are displayed" checkbox in the String Variable dialog box for the caption or message.

To remove or replace this feature	Do this
SHOWSTAR qualifier assigned to a caption or message	Clear the “Show * character if string is longer than field width” checkbox in the String Variable dialog box for the caption or message.
Embedded variable assigned to the title bar of an On Top display	Remove the embedded variable.
RSLinx® Enterprise™ Remote I/O alias with a BitArray data type	Assign a different data type to the alias.
Global object parameter definition or value	Delete the global object parameter definition. The corresponding value will be deleted automatically. Use parameter files instead.

Features that are not supported in version 3.20 or earlier

The features listed in the previous table are not supported in version 3.20 of RSView ME Station. In addition, these version 4.00 features are not supported in version 3.20 of RSView ME Station, nor in earlier versions. The right column describes how to remove or replace the feature.

To remove or replace this feature	Do this
Unsupported RSLinx Enterprise feature or shortcut—warning	Delete or replace the feature or shortcut. A hardware patch might be available that allows you to use the feature. Therefore, the runtime application file will still be created.
Unsupported RSLinx Enterprise feature or shortcut—error	Delete or replace the feature or shortcut.
Global reference object that is not linked to a global base object	Delete the global reference object, or link it to a global base object. All global reference objects that are linked to global base objects will be converted to standard graphic objects in the .mer file.
Language switch button graphic object	Delete the button.
Password button graphic object	Delete the button.
RecipePlus button graphic object	Delete the button.
RecipePlus selector graphic object	Delete the selector.

To remove or replace this feature	Do this
RecipePlus table graphic object	Delete the table.
Acknowledge all alarms button with a filtered trigger	Clear the Filtered triggers box.
Print alarm history button with a filtered trigger	Clear the Filtered triggers box.
Print alarm status button with a filtered trigger	Clear the Filtered triggers box.
Clear alarm history button with a filtered trigger	Clear the Filtered triggers box.
Clear alarm history button with the Reset alarm status option cleared	Select Reset alarm status.

Features that are not supported in version 3.10 or earlier

The features listed in the previous tables are not supported in version 3.10 of RSView ME Station. In addition, these version 3.20 alarm options are not supported in version 3.10, nor in earlier versions. The right column describes how to remove or replace the feature.

To remove or replace this feature	Do this
Alarm list graphic object with an unsupported combination of alarm conditions	Do one of the following: ■ Select the Display check box for each alarm condition ■ Select the Display check box for only these alarm conditions: ■ Active and unacknowledged ■ Inactive and unacknowledged.
Alarm list graphic object with Blink selected for one or more alarm conditions	For all alarm conditions that you are displaying, clear the Blink check box.
Alarm list graphic object with Use alarm colors turned off for one or more alarm conditions	For all alarm conditions that you are displaying, select the Use alarm colors check box.
Alarm list graphic object with the Acknowledged symbol column turned off	Select the Display column check box for the Acknowledged symbol column.

To remove or replace this feature	Do this
Alarm list graphic object with an Acknowledged symbol other than *	Change the Acknowledged symbol to *.
Alarm list graphic object with the Active symbol column displayed	Clear the Display column check box for the Active symbol column.
Alarm list graphic object with tags or expressions assigned to one or more connections	Clear the tags or expressions assigned to the connections.
Alarm list graphic object with Selected alarm indicator set to Cursor	Change the Selected alarm indicator to Highlight bar.
Alarm list graphic object with Lines per alarm set to >1	Change the Lines per alarm to 1.
Alarm banner graphic object with Use alarm colors selected	Clear the Use alarm colors check box.
Alarm status list graphic object with Use alarm colors selected	Clear the Use alarm colors check box.
Alarm status list graphic object with Lines per alarm set to >1	Change the Lines per alarm to 1.
Alarm status list graphic object with a Fore color other than white	Change the Fore color to white.

Features that are not supported in version 3.00

The features listed in the previous tables are not supported in version 3.00 of RSView ME Station. In addition, these version 3.10 features are not supported in version 3.00. The right column describes how to remove or replace the feature.

To remove or replace this feature	Do this
Piloted control list selector graphic object	Delete the object. Try using a control list selector instead.
Alarm banner graphic object with a filtered trigger	Clear the Filtered triggers box.
Alarm trigger with Message Notification connection assigned	Clear the tag or expression assigned to the connection.

To remove or replace this feature	Do this
Alarm trigger with Message Handshake connection assigned	Clear the tag or expression assigned to the connection.

APPENDIX H

Importing and exporting graphics XML files

This appendix describes:

- creating graphics XML files.
- exporting, editing, and importing XML files.
- the graphics XML file structure.

About XML

XML is the Extensible Markup Language used to create documents with structured text information. It has a standardized format and structure. You can use XML to edit the elements and attributes needed to create an alarm setup file or to modify graphic displays and global displays. For information about working with alarm XML files, see Appendix E.

Sometimes editing your display information in an XML file is quicker than working in FactoryTalk® View. For example, if you have a list of 100 local messages to set up for a graphic display, you might prefer to enter all the information in a text editor, and then import the display information into FactoryTalk View.

For more information about XML, see the World Wide Web Consortium's web page about XML at:
http://www.w3.org/XML.

Creating graphics XML files by exporting

The quickest way to create an XML file for your application's graphic displays is to export the data from FactoryTalk View. You can then open the XML file in Notepad, make your changes, and import the file back into FactoryTalk View.

The strings for the application's current language are exported to the XML file. To export strings for another language, reopen the application in the new language and repeat the XML export.

To export graphic display information to an XML file

1. In the Explorer window, right-click the Displays editor or the Global Objects editor.
2. Click Import and Export.

The Graphics Import Export Wizard opens.

3. Follow the instructions in the wizard.

   For information about using the Graphics Import Export Wizard, see Help.

FactoryTalk View creates XML files for the selected graphic displays, in the location you specify.

FactoryTalk View also creates a file called BatchImport_*Application name*.xml, in the same location. You can use this file to import multiple displays at the same time. To import a different set of displays than you exported, edit the list of display names in the BatchImport_*Application name*.xml file.

Editing XML files

We recommend that you use Notepad to edit your XML files.

If you do not want to change a property, you don't need to include it in the XML file. When you import the file, if you select the option "Create new objects in the display," properties that are not listed in the file are set to their default values. If you select the option "Update existing objects on the display," only properties that are listed in the file are updated with imported information.



If you include attributes for an object whose name does not match one of those in the graphic display, the attributes for that object are not imported. Attributes for all other objects in the file whose names do match the ones in the graphic display are imported.

Saving XML files in Notepad

Save XML files created or edited in Notepad using either UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to UTF-16 file format. For files containing strings in English or other Latin-based languages, UTF-8 is recommended, to reduce the size of the XML file. For other languages such as Chinese, Japanese, or Korean, UTF-16 is recommended.

The first line of every XML file contains XML version and encoding attributes. Make sure the encoding attribute matches the format that you are going to use when you save the file. For example, if the original file was saved in UTF-8 format and you plan to save it in UTF-16 format, make sure the first line specifies encoding="UTF-16".

Testing XML files

An XML file must be well-formed to be imported. To find out whether your XML file is well-formed, test it.

To test an XML file

1. Open the XML file in Internet Explorer.

If you can see the XML code, your file is well-formed. If the XML code is not well-formed, Internet Explorer displays an error message.

Importing XML files

You can import a graphic display or global object display that has been created using an external programming tool or editor, or you can import an XML file that you originally exported from FactoryTalk View and then modified.

When you import a graphic display or global object display, your existing display will be overwritten. Back up your application first, using the Application Manager tool. Or, you can save a copy of your existing display by exporting it to an XML file before you import the new one.

Error log file

If errors occur during importing, the errors are logged to a text file. The file opens automatically when importing is finished. The last paragraph of the file lists the location of the file.

Importing graphics XML files

You can import a single graphic or global object display XML file at a time, or import multiple displays. You can also choose whether to import new objects or update existing objects.

To import multiple displays, specify the names of the displays in the file BatchImport_*Application name*.xml. FactoryTalk View creates this file when you export multiple displays. For details, see page H-1.

To import display information from an XML file

1. In the Explorer window, right-click the Displays or Global Objects editor.
2. Click Import and Export.

   The Graphics Import Export Wizard opens.
3. Follow the instructions in the wizard.

   For more information about using the Graphics Import Export Wizard, see Help.

Graphics XML file structure

The graphic display or global object display XML file is a FactoryTalk View XML document that describes the objects and settings for a display. The root element of the XML document is called **gfx**. It represents the display. An XML document can contain only one root element. All other elements in the document must be contained or nested within the root element.

In an XML document, the start of an element is marked **<element name>**. The end is marked **</element name>**.

If the element contains no subelements, the end can be marked />. For example, <caption fontFamily="Arial" fontSize="8" bold="false" />.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes.

Here is a sample structure for a graphic display XML document containing two graphic objects. The second object has states:

Element	Description
<gfx>	Root element.
<displaySettings />	Contains attributes from the Display Settings dialog box in the Graphics editor.
<*object1*>	Contains attributes from the General and Common tabs in the object's Properties dialog box, as well as elements for the object's caption, image, animation, and connections.
<caption />	Contains attributes for the object's caption.
<imageSettings />	Contains attributes for the object's image.
<animations>	Contains an animation element for each type of animation set up for the object.
<animateVisibility />	Contains attributes for Visibility animation.
<animateColor />	Contains attributes for Color animation.
</animations>	Indicates the end of the animations element.
<connections>	Contains a connection element for each connection assigned to the object.
<connection name= "Value" />	Contains attributes for the Value connection.
<connection name= "Indicator" />	Contains attributes for the Indicator connection.
</connections>	Indicates the end of the connections element.
</*object1*>	Indicates the end of the *object1* element.
<*object2*>	Contains attributes from the General and Common tabs in the object's Properties dialog box, as well as elements for the object's states and connections.
<states>	Contains state elements for each of the object's states.

Element	Description
<state stateid="0">	Contains attributes for the object's first state, as well as elements for the state's caption and image.
<caption />	Contains attributes for the state's caption.
<imageSettings />	Contains attributes for the state's image.
</state>	Indicates the end of the state element.
<state stateid="1">	Contains attributes for the object's second state, as well as elements for the state's caption and image.
<caption />	Contains attributes for the state's caption.
<imageSettings />	Contains attributes for the state's image.
</state>	Indicates the end of the state element.
</states>	Indicates the end of the states element.
<connections>	Contains a connection element for each connection assigned to the object.
<connection name= "Value" />	Contains attributes for the Value connection.
<connection name= "Indicator" />	Contains attributes for the Indicator connection.
</connections>	Indicates the end of the connections element.
</*object2*>	Indicates the end of the *object2* element.
</gfx>	Indicates the end of the gfx element.

You can specify multiple attributes for an element. For example, the caption element contains 13 possible attributes.

Elements for group objects begin with **<group>** and end with **</group>**. The <group> element contains all the elements for each object in the group.

For more information about graphic object elements and their attributes, see Help for the Graphics Import Export Wizard.

Index

Symbols

.bmp files ▪ 19-20
  importing ▪ 19-21
  location of ▪ 19-21
  placing in graphic displays ▪ 20-14
  tips for using ▪ 19-24
  viewing ▪ 19-22
.dxf files ▪ 19-20
  placing in graphic displays ▪ 20-20
.jpg files ▪ 19-20
  importing ▪ 19-21
  placing in graphic displays ▪ 20-14
  tips for using ▪ 19-24
  viewing ▪ 19-22
.mea files ▪ 4-11
.med files ▪ 4-2
.mer files ▪ 4-2
  comparing ▪ 16-7
  converting to .med ▪ 14-3, 14-6
  creating ▪ 14-2
  transferring
    from PanelView Plus or PanelView Plus CE ▪ 16-6
    to PanelView ▪ 16-1, 16-5
    to PanelView Plus CE ▪ 16-1, 16-5
    to personal computers ▪ 15-1, 15-3
.wmf files ▪ 19-20
  placing in graphic displays ▪ 20-20
.xml files ▪ 9-4, 19-4, E-1, H-1

Numerics

21 CFR Part 11 compliance ▪ 10-4

A

A.I. 5 tags
  importing ▪ 7-11
AB_DF1-1
  using to transfer applications ▪ 16-5
AB_ETH-1
  using to transfer applications ▪ 16-5
Acknowledge alarm button graphic object ▪ 20-7, 21-16
  using with alarm objects ▪ 9-8, 9-35
Acknowledge all alarms button graphic object ▪ 9-8, 20-8, 21-16
  using with alarm lists ▪ 9-36
ActiveX objects ▪ 20-2
  animating ▪ 22-2
  at run time ▪ 17-17, 17-18
  creating ▪ 20-21
  navigating to ▪ 21-7
  using to display tag values ▪ 20-6, 20-9
  using to set tag values ▪ 20-5, 20-9
Addressing syntax
  for HMI tags ▪ 7-5
  Logix5000 ▪ 7-6
Adobe Reader ▪ 1-1
ALARM BANNER display
  editing ▪ 9-26
Alarm banner graphic object ▪ 20-10, 21-59
  at run time ▪ 9-33, 17-19
  linking buttons to ▪ 21-9
  navigating to ▪ 21-7
  specifying time and date format for ▪ 15-16
  using buttons with ▪ 9-35
ALARM display ▪ 9-6
  editing ▪ 9-25
  window size ▪ 4-14
Alarm display
  opening and closing ▪ 9-25, 9-31
    remotely ▪ 9-25
Alarm Import Export Wizard ▪ E-1
Alarm list graphic object ▪ 20-10, 21-58
  at run time ▪ 9-32, 17-19
  linking buttons to ▪ 21-9
  navigating to ▪ 21-7
  specifying time and date format for ▪ 15-16
  using buttons with ▪ 9-35
Alarm log file ▪ 4-3, 9-10
  deleting at application startup ▪ 15-15
  printing contents of ▪ 9-10

Alarm messages
embedded variables in ▪ 24-2
exporting to XML ▪ E-1
handshaking ▪ 9-21
importing XML ▪ E-3
multiple languages for ▪ 9-5, 9-17
printing at run time ▪ 9-10
viewing at run time ▪ 17-19
ALARM MULTI-LINE display ▪ 9-6
copying ▪ 9-30
editing ▪ 9-27
Alarm Setup editor ▪ 9-3
exporting to XML ▪ E-1
importing XML ▪ E-3
using tags in ▪ 6-9
Alarm status list graphic object ▪ 20-10, 21-60
at run time ▪ 9-34
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using buttons with ▪ 9-35
Alarm status mode button graphic object ▪ 20-7, 21-16
using with alarm status lists ▪ 9-36
Alarm triggers ▪ 9-4
and the default language ▪ 9-5
data types for ▪ 9-11
exporting to XML ▪ E-1
importing XML ▪ E-3
Alarms ▪ 9-1
acknowledging ▪ 9-8, 9-19, 9-38
remotely ▪ 9-19, 9-20, 9-24
clearing ▪ 9-8, 9-39
creating your own display for ▪ 9-30
deleting ▪ 9-39
displaying ▪ 9-6
displaying in the alarm status list ▪ 9-40
expressions in ▪ 23-5
filtering ▪ 9-5
handshaking ▪ 9-18
for remote acknowledgements ▪ 9-20
importing and exporting ▪ 9-4, 19-4
keeping a permanent record of ▪ 9-6, 9-10
notification methods for ▪ 9-5
planning ▪ 3-5
printing ▪ 9-10
queueing ▪ 9-18
resetting status of ▪ 9-9, 9-24, 9-40
remotely ▪ 9-25
responding to ▪ 9-7
retaining status of ▪ 9-40
sending messages to the data source ▪ 9-22
setting up ▪ 9-2
silencing ▪ 9-8, 9-24, 9-39
remotely ▪ 9-24
sorting ▪ 9-9, 9-40
testing on the development computer ▪ 14-2
Analog tags ▪ 7-1
how floating-point values are rounded ▪ 7-2
logging values for ▪ 26-5
monitoring for alarms ▪ 9-4
using to generate a range of alarms ▪ 9-11
Animation ▪ 22-1
checking ▪ 22-15
copying ▪ 22-16
defining range of motion for ▪ 22-6
for global objects ▪ 22-17, 25-9
for group objects ▪ 20-47, 22-14
setting minimum and maximum values for ▪ 22-6
testing ▪ 22-5
using expressions ▪ 22-6
using Object Smart Path ▪ 22-4, 22-7
using tags and placeholders ▪ 22-5
viewing in Object Explorer ▪ 20-25
Animation dialog box ▪ 22-2
Animation types ▪ 22-1
color ▪ 22-9
fill ▪ 22-12
height ▪ 22-13
horizontal position ▪ 22-12
horizontal slider ▪ 22-14
rotation ▪ 22-13
vertical position ▪ 22-13
vertical slider ▪ 22-14
visibility ▪ 22-8
width ▪ 22-13
Application Explorer ▪ 2-4, 2-5
showing and hiding ▪ 2-7
using ▪ 2-8
Application files
converting from runtime to development ▪ 14-3, 14-6

deleting from disk ▪ 18-4
viewing location of ▪ 4-18
Application Manager ▪ 4-10
converting .mer files to .med files ▪ 14-6
security for ▪ 4-10
Application properties
viewing ▪ 4-18
Application text
exporting for translation ▪ 12-6
importing ▪ 12-13
translating in Excel ▪ 12-9
translating in Unicode ▪ 12-9
Applications ▪ 2-1, 18-1
backing up and restoring ▪ 4-10
closing ▪ 4-10
comparing ▪ 16-7
converting
PanelBuilder ▪ B-1
PanelBuilder 1400e ▪ A-1
PanelBuilder 32 ▪ B-1
copying, deleting, and renaming ▪ 4-10
creating ▪ 4-4
importing
PanelBuilder ▪ B-1
PanelBuilder 1400e ▪ A-1
PanelBuilder 32 ▪ B-1
multiple version support for ▪ 4-10, G-1
opening ▪ 4-7
multiple ▪ 4-9
problems with ▪ 4-9
planning ▪ 3-1
runtime, creating ▪ 14-2
sample, granting users access to ▪ 2-2
sample, opening ▪ 2-2
security for ▪ 11-18, 11-19, 11-20
specifying language for ▪ 4-5, A-4, B-3
testing on the development computer ▪ 14-2
transferring from PanelView Plus or PanelView Plus CE ▪ 16-6
transferring to PanelView Plus or PanelView Plus CE ▪ 16-5
using earlier versions of FactoryTalk View ▪ G-1
using earlier versions of RSView ▪ 4-10
versus projects ▪ 4-1
Arc graphic object ▪ 20-2, 20-16
Arithmetic operators
in expressions ▪ 23-7
evaluation order of ▪ 23-11
Array tags ▪ 9-15
data types for ▪ 9-15, 9-16
Arrow images ▪ 19-20
Audiences
for diagnostics messages ▪ 10-4
Auto repeat ▪ 21-12
AutoCAD files
See .dxf files
Automatic logout ▪ 11-13, 11-23

B

Background Color toolbar ▪ 20-32
Background style
transparent ▪ 19-21
Backing tags ▪ 6-6, 25-13
using with global object parameters ▪ 25-11
Backspace button graphic object ▪ 20-7, 21-16, 21-48
Bar graph graphic object ▪ 20-6, 21-43, 21-45
at run time ▪ 17-18
using animation to create ▪ 22-10
using with recipes ▪ 29-10
Base objects
assigning global object parameters to ▪ 25-11, 25-12
breaking links to ▪ 25-10
creating ▪ 25-7
deleting ▪ 25-10
editing ▪ 25-9
Bit arrays
using to generate alarms based on priority sequence ▪ 9-13
using to generate multiple alarms ▪ 9-12
Bit trigger type ▪ 9-12
Bitmap files
See .bmp files
Bitwise operators
in expressions ▪ 23-9
evaluation order of ▪ 23-11
Bounding box ▪ 21-5
Button graphic objects
bounding box ▪ 21-5

creating touch margins for ▪ 21-4
linking to specific objects ▪ 21-9
repeating button presses ▪ 21-12
selecting objects to send button presses to ▪ 21-7, 21-9
using function keys with ▪ 21-5
using with alarm banners ▪ 20-7, 20-8
using with alarm history and diagnostics messages ▪ 20-8
using with lists ▪ 20-7
using with numeric input objects ▪ 20-7, 20-8
using with trends ▪ 20-7, 20-8, 28-11
Button presses
repeating ▪ 21-12
sending to graphic objects ▪ 21-7, 21-9

C

Caches for data servers
synchronizing ▪ 5-5
Change Password window ▪ 17-5
Circle graphic object ▪ 20-2, 20-17
Clear alarm banner button graphic object ▪ 9-8, 20-7, 21-16
using with alarm banners ▪ 9-36
Clear alarm history button graphic object ▪ 20-8, 21-17
clearing alarms with ▪ 9-8
resetting alarms with ▪ 9-9
silencing alarms with ▪ 9-9
using to delete the alarm log file ▪ 9-10
using with alarm lists ▪ 9-36
Close display button graphic object ▪ 20-4, 21-36
and alarms ▪ 9-31
and diagnostics messages ▪ 10-11
and information messages ▪ 27-6, 27-7
using to navigate ▪ 13-3, 13-6
Color
blinking ▪ C-1
for graphic objects ▪ 20-32
in graphic displays
animating ▪ 22-9
in trends ▪ 28-9
Color animation ▪ 20-6, 22-9
at run time ▪ 17-18
Communication drivers
setting up at run time
for transfer to PanelView Plus ▪ 16-3
for transfer to PanelView Plus CE ▪ 16-3
on personal computers ▪ 15-9
setting up in RSLinx Enterprise
at run time ▪ 15-9
for transfer to PanelView Plus ▪ 16-5
for transfer to PanelView Plus CE ▪ 16-5
Communications
displaying errors at run time ▪ 17-20
setting up ▪ 5-1
testing on the development computer ▪ 14-2
Components ▪ 2-11, 18-1
adding into applications ▪ 18-3
file locations ▪ 4-2
printing ▪ 2-13, 18-5
security for ▪ 11-18
working with ▪ 18-1
Connections ▪ 20-2
assigning to graphic objects ▪ 20-36
using the Property Panel ▪ 20-31
expressions ▪ 20-2
for alarms ▪ 9-4, 9-17, 9-23
tags ▪ 20-2
Constants
in expressions ▪ 23-6
Context menus ▪ 19-5
Control list selector graphic object ▪ 20-5, 21-47
at run time ▪ 17-17
differences from piloted control list selectors ▪ 21-50
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using Enter key handshaking with ▪ 21-13
Current trends ▪ 26-6, 28-2

D

Data log files ▪ 4-3, 26-1
Data log models
changing the model to use at run time ▪ 26-5
deleting tags from ▪ 26-5

Data Log Models editor ▪ 26-2
using tags in ▪ 6-12
Data logging ▪ 26-1
choosing the data to log ▪ 26-5
methods ▪ 26-4
problems with ▪ 26-6
setting up ▪ 26-1
storage locations ▪ 26-3
to monitor memory usage ▪ 8-8
Data server tags ▪ 6-1
browsing for ▪ 6-5
how to use ▪ 6-3
when to use ▪ 6-3
Data servers ▪ 2-8, 4-1, 5-2
creating ▪ 5-4
synchronizing caches for ▪ 5-5
Data sets, for recipes ▪ 29-1
comparing ▪ 29-7
printing ▪ 29-9
Data source
ensuring values are read by ▪ 21-13
for alarms ▪ 9-3
for HMI tags ▪ 7-5
for information messages ▪ 27-3
for local messages ▪ 19-27
for tags ▪ 6-2, 7-5
for trends ▪ 28-2
Default graphic displays
ALARM display ▪ 9-6
DIAGNOSTICS display ▪ 10-10
INFORMATION display ▪ 27-6
Default language ▪ 12-2
and alarm triggers ▪ 9-5
DEFAULT user ▪ 11-4
at runtime ▪ 11-3
DeskLock tool ▪ 11-2, 15-17
Device shortcuts
editing at run time ▪ 15-6
Device tags ▪ 7-5
addressing syntax for ▪ 7-5
Diagnostics clear all button graphic object ▪ 20-8, 21-17
using with diagnostic lists ▪ 10-12
Diagnostics clear button graphic object ▪ 20-7, 21-17
using with diagnostics lists ▪ 10-12
DIAGNOSTICS display ▪ 10-10
at run time ▪ 17-19
opening and closing ▪ 10-11
window size ▪ 4-14
Diagnostics List ▪ 2-5
clearing messages in ▪ 2-5
displaying ▪ 10-5
moving ▪ 2-5
resizing ▪ 2-5
showing and hiding ▪ 2-7
using to test displays ▪ 19-10
Diagnostics list graphic object ▪ 20-10, 21-62
at run time ▪ 10-11, 17-19
displaying communication errors in ▪ 17-20
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using buttons with ▪ 10-12
Diagnostics List Setup editor ▪ 10-9
Diagnostics messages ▪ 10-1
audiences for ▪ 10-4
creating your own display for ▪ 10-11
displaying at run time ▪ 10-8, 17-19
displaying during application development ▪ 10-4
logging from PanelView Plus and PanelView Plus CE terminals ▪ 10-6, 10-8
logging to an ODBC database ▪ 10-7
message severity ▪ 10-3
ODBC format for ▪ D-1
printing at run time ▪ 10-8
setting up at run time ▪ 10-10
setting up destinations for ▪ 10-6
setting up message routing for ▪ 10-6, 10-8
viewing log files ▪ 10-5
Diagnostics Setup tool ▪ 10-6
Diagnostics Viewer tool ▪ 10-5
Digital tags ▪ 7-1
logging values for ▪ 26-5
monitoring for alarms ▪ 9-4
DIN symbols ▪ 19-20
Display editor
See Graphic Displays editor

Display list selector graphic object ▪ 20-4, 21-37
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using parameter files with ▪ 25-2
using to navigate ▪ 13-3, 13-6
Display number
specifying ▪ 8-4, 8-5
Display print button graphic object ▪ 19-30, 20-4, 21-17
printing trend data with ▪ 28-12
using the data source instead of ▪ 8-5
Display Settings dialog box ▪ 19-11
exporting to XML ▪ H-1
importing XML ▪ H-3
Display size
See Window size
Display types ▪ 19-12
and navigation ▪ 13-4
default ▪ 19-3
Displays
See Graphic displays
Documentation for FactoryTalk View
finding ▪ P-2
printing ▪ 1-1
Drawing objects ▪ 20-1
animating ▪ 22-2
coloring ▪ 20-33
creating ▪ 20-12
flipping ▪ 20-54
locking into position ▪ 20-56
reshaping ▪ 20-45
rotating ▪ 20-55
Drivers
See Communication drivers

E

Edit mode ▪ 19-10
Editor types
Alarm Setup ▪ 9-3
Data Log Models ▪ 26-2
Diagnostics List Setup ▪ 10-9
Expression ▪ 23-2
Global Connections ▪ 8-2
Graphic Displays ▪ 19-3
Graphic Libraries ▪ 19-15
Images ▪ 19-22
Information Messages ▪ 27-2
Information Setup ▪ 27-3
Local Messages ▪ 19-26
Parameters ▪ 25-3
Project Settings ▪ 4-17
RecipePlus Editor ▪ 29-6, 29-7
RecipePlus Setup ▪ 29-5
Runtime Security ▪ 11-3
Editors
printing in ▪ 2-13, 18-5
working with ▪ 2-10
Ellipse graphic object ▪ 20-2, 20-17
Embedded variables ▪ 24-1
displaying at run time ▪ 24-7
in title bars ▪ 19-13
syntax for ▪ 24-3
types of values ▪ 24-1
updating at run time ▪ 24-6
End button graphic object ▪ 20-7, 21-17, 21-48
using with alarm objects ▪ 9-38
using with diagnostics lists ▪ 10-12
using with recipes ▪ 29-12
using with trends ▪ 28-12
Enter button graphic object ▪ 20-7, 21-17, 21-48
and Enter key handshaking ▪ 21-13
using with alarm objects ▪ 9-8, 9-38
using with recipes ▪ 29-12
Enter key handshaking ▪ 21-13
Error messages
See
Diagnostics messages
Problems at run time
Exporting
alarm files ▪ 9-4, E-1
global object displays ▪ H-1
graphic displays ▪ 19-4, H-1
problems with ▪ 12-8
text for translation ▪ 12-6
Expression editor ▪ 23-2
using tags in ▪ 6-5
Expression results
rounding ▪ 23-1
types of values ▪ 23-2

Expressions ▪ 23-1
and animation ▪ 22-6
assigning to connections ▪ 6-11
assigning to graphic objects ▪ 20-36
using the Property Panel ▪ 20-31
constants in ▪ 23-6
evaluation order of operators ▪ 23-11
formatting ▪ 23-5
if-then-else logic in ▪ 23-17
language function in ▪ 23-15
math functions in ▪ 23-14
monitoring for alarms ▪ 9-4
operators in
arithmetic ▪ 23-7
bitwise ▪ 23-9
logical ▪ 23-8
relational ▪ 23-8
security functions in ▪ 23-15
tags and placeholders in ▪ 23-5
viewing in Object Explorer ▪ 20-25
write expressions ▪ 23-19

F

Faceplates
See Process faceplates
Factory Talk Local Directory
restoring from runtime computer to development computer ▪ 14-6
FactoryTalk
loading directory of users and security policies ▪ 15-5
logging in ▪ 11-3
at runtime ▪ 11-3
single sign-on ▪ 11-22
turning off warning ▪ 15-16
FactoryTalk Diagnostics ▪ 10-1
setting up
on a personal computer ▪ 15-7
See also Diagnostics messages
FactoryTalk Diagnostics Viewer tool ▪ 10-5
viewing diagnostics log files in ▪ 10-5
Windows Event Viewer ▪ 10-5
FactoryTalk Directory
security access to ▪ 11-19
FactoryTalk Security ▪ 11-1
permissions for converting runtime applications to development applications ▪ 14-6
setting up ▪ 11-14
turning off ▪ 11-2
users, creating ▪ 11-17
FactoryTalk View
FactoryTalk View ME Station ▪ 1-1
FactoryTalk View Studio ▪ 1-1
FactoryTalk View Machine Edition ▪ 1-1
FactoryTalk View ME Station
exiting ▪ 15-5
starting
on PanelView Plus CE terminals ▪ 16-3
on PanelView Plus terminals ▪ 16-3
on personal computers ▪ 15-4, 15-11
supported versions ▪ 4-10, G-1
FactoryTalk View Studio
main window ▪ 2-4
starting and exiting ▪ 2-1
FactoryTalk View tags
importing ▪ 7-11
File names ▪ 4-2
and spaces ▪ 4-4
maximum length of ▪ 4-4
Fill animation ▪ 20-6, 22-12
at run time ▪ 17-18
Folders ▪ 7-7
Fonts
for multiple languages ▪ 20-13
in graphic displays
substitution at run time ▪ 20-13
in trends ▪ 28-9
transferring ▪ 16-5
Foreground Color toolbar ▪ 20-32
Freehand graphic object ▪ 20-2, 20-17
Function keys
assigning to graphic objects ▪ 21-5

G

Gauge graphic object ▪ 20-6, 21-43, 21-46
at run time ▪ 17-18
specifying number format for ▪ 15-16
Global connections ▪ 8-1
setting up ▪ 8-2

using to change displays remotely ▪ 8-4, 13-7
using to monitor memory usage ▪ 8-7
using to print displays remotely ▪ 8-5
using to run macros ▪ 8-6, 30-3
Global Connections editor ▪ 8-2, 19-30
Global object displays
creating ▪ 25-7
exporting to XML ▪ H-1
importing XML ▪ H-3
Global object parameters ▪ 25-1
setting up ▪ 25-10
specifying tags for ▪ 25-13
using with process faceplates ▪ 25-11
Global objects ▪ 25-1
animating ▪ 22-17, 25-9
creating ▪ 25-7, 25-8
exporting to XML ▪ H-1
importing XML ▪ H-3
using global object parameters with ▪ 25-10
See also
Base objects
Reference objects
Goto configure mode button graphic object ▪ 20-4, 21-17
using to navigate ▪ 13-3
Goto display button graphic object ▪ 20-4, 21-35
and alarms ▪ 9-30, 9-31
and diagnostics messages ▪ 10-11
and information messages ▪ 27-7
assigning visibility animation to ▪ 11-24
using parameter files with ▪ 25-2
using to navigate ▪ 13-3, 13-4
Graphic displays ▪ 19-1
changing remotely ▪ 8-4, 13-7
security issues ▪ 8-4
setting up ▪ 8-4
using global connections ▪ 8-4
when a user logs out ▪ 11-23
creating ▪ 19-3
creating a background for ▪ 19-14
default
ALARM ▪ 9-6
DIAGNOSTICS ▪ 10-10
INFORMATION ▪ 27-6
developing a hierarchy for ▪ 13-1
exporting to XML ▪ H-1
illustrating ▪ 20-2
importing XML ▪ H-3
limiting access to ▪ 11-9
maximum number licensed to use ▪ 4-18
navigating between ▪ 13-1
objects for ▪ 13-3
testing navigation ▪ 13-3
numbering ▪ 8-4, 8-5
opening ▪ 19-3
planning ▪ 3-3, 19-2
printing ▪ 19-29
at run time ▪ 19-30
remotely ▪ 8-5, 19-30
setting up printer ▪ 15-10
setting up ▪ 19-11
sizing ▪ 4-12, 19-13
testing ▪ 19-10
animation ▪ 22-5
title bar in ▪ 4-15
types of ▪ 19-12
default type ▪ 19-3
using parameter files with ▪ 25-2
viewing in detail ▪ 19-9
Graphic Displays editor ▪ 19-3
undoing and redoing actions in ▪ 19-10
using tags in ▪ 6-9
Graphic images
importing ▪ 19-20
monochrome ▪ 19-21
that come with FactoryTalk View Studio ▪ 19-21
viewing ▪ 19-22
Graphic libraries ▪ 19-15
ALARM BANNER display ▪ 9-26
ALARM MULTI-LINE display ▪ 9-6, 9-26, 9-27
changing file location for ▪ 19-19
copying ▪ 9-30
creating ▪ 19-15
HISTORY display ▪ 9-6, 9-29
opening ▪ 19-15
RecipePlus_Components library ▪ 29-10
saving with multiple languages ▪ 19-17
setting up for multiple languages ▪ 12-14

Sliders library ▪ 22-14
STATUS display ▪ 9-6, 9-28
Trends library ▪ 28-10
Graphic Libraries editor ▪ 19-15
Graphic object types
acknowledge alarm button ▪ 20-7, 21-16
acknowledge all alarms button ▪ 20-8, 21-16
ActiveX objects ▪ 20-2, 20-21
alarm banner ▪ 20-10, 21-59
alarm list ▪ 20-10, 21-58
alarm status list ▪ 20-10, 21-60
alarm status mode button ▪ 20-7, 21-16
backspace button ▪ 20-7, 21-16, 21-48
bar graph ▪ 20-6, 21-43, 21-45
clear alarm banner button ▪ 20-7, 21-16
clear alarm history button ▪ 20-8, 21-17
close display button ▪ 20-4, 21-36
control list selector ▪ 20-5, 21-47
diagnostics clear all button ▪ 20-8, 21-17
diagnostics clear button ▪ 20-7, 21-17
diagnostics list ▪ 20-10, 21-62
display list selector ▪ 20-4, 21-37
display print button ▪ 20-4, 21-17
drawing objects ▪ 20-1
arcs ▪ 20-2, 20-16
circles ▪ 20-2, 20-17
ellipses ▪ 20-2, 20-17
freehand objects ▪ 20-2, 20-17
images ▪ 20-2, 20-14
lines ▪ 20-2, 20-18
panels ▪ 20-2, 20-16
polygons ▪ 20-2, 20-18
polylines ▪ 20-3, 20-18
rectangles ▪ 20-3, 20-19
rounded rectangles ▪ 20-3, 20-20
squares ▪ 20-3, 20-19
text ▪ 20-2, 20-13
wedges ▪ 20-3, 20-16
end button ▪ 20-7, 21-17, 21-48
enter button ▪ 20-7, 21-17, 21-48
gauge ▪ 20-6, 21-43, 21-46
goto configure mode button ▪ 20-4, 21-17
goto display button ▪ 20-4, 21-35
home button ▪ 20-7, 21-17, 21-48
information acknowledge button ▪ 20-7, 21-17
information message display ▪ 20-10, 21-63
interlocked push button ▪ 20-5, 21-26
language switch button ▪ 20-4, 21-17
latched push button ▪ 20-4, 21-23
list indicator ▪ 20-6, 21-42
local message display ▪ 20-10, 21-53
login button ▪ 20-4, 21-17
logout button ▪ 20-4, 21-16
macro button ▪ 21-54
maintained push button ▪ 20-4, 21-21
momentary push button ▪ 20-4, 21-20
move down button ▪ 20-7, 21-16, 21-48
move left button ▪ 20-7, 21-16
move right button ▪ 20-7, 21-16
move up button ▪ 20-7, 21-16, 21-48
multistate indicator ▪ 20-6, 21-39
multistate push button ▪ 20-5, 21-24
next pen button ▪ 20-7, 21-17
numeric display ▪ 20-9, 21-28
numeric input cursor point ▪ 20-9, 21-32
numeric input enable button ▪ 20-9, 21-30
page down button ▪ 20-7, 21-17, 21-48
page up button ▪ 20-7, 21-17, 21-48
password button ▪ 20-4, 21-17
pause button ▪ 20-7, 21-17
piloted control list selector ▪ 20-5, 21-50
print alarm history button ▪ 20-4, 21-56
print alarm status button ▪ 20-4, 21-57
ramp button ▪ 20-5, 21-27
RecipePlus button ▪ 20-9, 21-17, 29-3
RecipePlus selector ▪ 20-9, 29-2
RecipePlus table ▪ 20-9, 29-2
reset alarm status button ▪ 21-17
reset alarm status mode button ▪ 20-8
return to display button ▪ 20-4, 21-17
scale ▪ 20-6, 21-43, 21-47
shutdown button ▪ 20-4, 21-17
silence alarms button ▪ 20-8, 21-17
sort alarms button ▪ 20-8, 21-17
string display ▪ 20-9, 21-33
string input enable button ▪ 20-9, 21-34
symbol ▪ 20-6, 21-40
time and date display ▪ 20-6, 21-55
trend ▪ 20-6

Graphic objects ▪ 19-1, 20-1
aligning ▪ 20-50
animating ▪ 22-2
arranging ▪ 20-49, 20-53
assigning connections to ▪ 20-36
using the Property Panel ▪ 20-31
coloring ▪ 20-32
converting to wallpaper ▪ 19-14
copying ▪ 20-42
creating ▪ 20-12
selecting tools for ▪ 20-10
deleting ▪ 20-46
deselecting ▪ 20-23
duplicating ▪ 20-43
editing ▪ 20-40
embedded variables in ▪ 24-2
exporting to XML ▪ H-1
grouping and ungrouping ▪ 20-47
importing XML ▪ H-3
moving ▪ 20-41
naming ▪ 20-34
navigating between ▪ 21-7
positioning ▪ 20-49
with grid ▪ 19-8
replacing tags and expressions in ▪ 20-38
resizing ▪ 20-44
selecting ▪ 20-23
setting up ▪ 21-1
using the Properties dialog box ▪ 20-26
using the Property Panel ▪ 20-30
spatial properties of ▪ 21-1
time, date, and number formats ▪ 21-16
using at run time ▪ 21-3
using keys to work with *See* Special keys
using to display alarms and messages ▪ 20-10
using to display processes ▪ 20-6
using to enter and display numeric and string values ▪ 20-9
using to illustrate displays ▪ 20-2
using to navigate ▪ 13-3
using to start and control processes ▪ 20-4
visibility, setting up ▪ 21-1
Grid
in graphic displays ▪ 19-8
in trends ▪ 28-9
Group objects
animating ▪ 20-47, 22-14
creating ▪ 20-47
editing ▪ 20-48
naming ▪ 20-35
using global object parameters with ▪ 25-11

H

Handshaking
Enter key ▪ 21-13
for Alarm messages ▪ 9-21
for alarms ▪ 9-18
remote ▪ 9-20
Height animation ▪ 20-6, 22-13
at run time ▪ 17-18
Historical trends ▪ 26-6, 28-2
HISTORY display ▪ 9-6
editing ▪ 9-29
HMI servers ▪ 2-8, 2-9, 4-1
HMI tags ▪ 6-2, 7-1
addressing syntax for ▪ 7-5
browsing for ▪ 6-5
creating ▪ 7-8
data sources ▪ 7-5
exporting ▪ 7-11
how to use ▪ 6-5
importing ▪ 7-9, 7-11
merging databases ▪ 7-11
naming ▪ 7-7
searching for ▪ 7-4
when to use ▪ 6-4
See also
Analog tags
Device tags
Digital tags
Memory tags
String tags
System tags
Home button graphic object ▪ 20-7, 21-17, 21-48
using with alarm objects ▪ 9-37
using with diagnostics lists ▪ 10-12
using with recipes ▪ 29-12
using with trends ▪ 28-11

Horizontal position animation ■ 20-6, 22-12
at run time ■ 17-18
Horizontal slider animation ■ 20-6, 22-7, 22-14
at run time ■ 17-17

I

If-then-else logic
in expressions ■ 23-17
nesting ■ 23-18
Image Browser ■ 19-23
Image graphic object ■ 20-2, 20-14
Images editor ■ 19-22
Importing
alarm files ■ 9-4, E-3
applications ■ 4-6
global object displays ■ H-3
graphic displays ■ 19-4, H-3
tags ■ 7-11
Importing application text ■ 12-13
problems with ■ 12-13
Indicators ■ 21-38
Information acknowledge button graphic object ■ 20-7, 21-17
at run time ■ 27-7
INFORMATION display ■ 27-6
at run time ■ 17-19
opening and closing ■ 27-7
window size ■ 4-14
Information message display graphic object ■ 20-10, 21-63
at run time ■ 17-19, 27-7
Information message files ■ 27-2, 27-4
changing the file to use at run time ■ 27-8
Information messages ■ 27-1
creating your own display for ■ 27-7
displaying ■ 27-3
embedded variables in ■ 24-2
multiple languages for ■ 27-5
setting up ■ 27-1
viewing at run time ■ 17-19
Information Messages editor ■ 27-2
Information Setup editor ■ 27-3
Ingredients, for recipes ■ 29-1
Initial values for tags ■ 6-12, 30-1
See also Tag values
Input focus
giving to graphic objects ■ 21-7
Interlocked push button graphic object ■ 20-5, 21-26
at run time ■ 17-16
ISA symbols ■ 19-20
Isolated graphing
in trends ■ 28-8

K

KEPServerEnterprise
array tag syntax ■ 9-17
Keyboard button symbols ■ 19-21
Keyboard navigation ■ 21-7

L

Language function
in expressions ■ 23-15
Language strings
exporting for translation ■ 12-6
importing ■ 12-13
translating in Excel ■ 12-9
translating in Unicode ■ 12-9
Language switch button graphic object ■ 12-3, 20-4, 21-17
placing in displays ■ 13-3
planning ■ 3-5
Language switching ■ 12-1
See also Languages
setting up ■ 12-3
setting up Windows for ■ 12-4
Languages
adding ■ 12-5
to libraries ■ 19-17
choosing fonts for ■ 20-13
default ■ 12-2
displaying current language ■ 23-15
planning ■ 3-5
display navigation ■ 13-3
RFC1766 names ■ F-1

setting up messages
alarm ▪ 9-5, 9-17
for RSView ME Station 4.00 ▪ 23-16
information ▪ 27-5
local ▪ 19-29
testing ▪ 14-2
Latched push button graphic object ▪ 20-4, 21-23
at run time ▪ 17-16
Least Significant Bit trigger type ▪ 9-13
Libraries
See Graphic libraries
License
viewing maximum allowable graphic displays ▪ 4-18
Line graphic object ▪ 20-2, 20-18
Link properties
for reference objects ▪ 25-9
List indicator graphic object ▪ 20-6, 21-42
at run time ▪ 17-18
Local message display graphic object ▪ 19-27, 20-10, 21-53
at run time ▪ 17-19, 19-29
using with message files ▪ 19-25
Local message files ▪ 19-26, 19-27
Local messages ▪ 19-25
embedded variables in ▪ 24-2
multiple languages for ▪ 19-29
viewing at run time ▪ 17-19
Local Messages editor ▪ 19-26
Localizing applications
importing text ▪ 12-13
information for translators ▪ 12-9
new line character ▪ 12-12
problems importing text ▪ 12-13
Unicode file schema ▪ 12-11
Log files
ODBC format for diagnostics messages ▪ D-1
LOGIC 5
See WINtelligent LOGIC 5 tags
Logical operators
in expressions ▪ 23-8
evaluation order of ▪ 23-11
Login button graphic object ▪ 20-4, 21-17
locating in display hierarchy ▪ 13-1
logging in with ▪ 11-12
Login macro ▪ 30-2, 30-4
assigning to RSView 3.20 and earlier users ▪ 11-6
assigning to users ▪ 11-4
running ▪ 17-3, 17-6
Login window ▪ 17-2
Logix5000 faceplates
See Process faceplates
Logix5000 processors ▪ 6-2
addressing syntax for ▪ 7-6
creating tags in ▪ 6-3
Logout button graphic object ▪ 20-4, 21-16
locating in display hierarchy ▪ 13-1
logging out with ▪ 11-12
Logout macro ▪ 30-2, 30-4
assigning to RSView 3.20 and earlier users ▪ 11-6
assigning to users ▪ 11-4
running ▪ 17-3, 17-6
LSBit trigger type ▪ 9-13

M

Macro button graphic object ▪ 21-54
using to run macros ▪ 30-2
Macros
creating ▪ 30-1
remote
running from the data source ▪ 8-6, 30-3
Macros editor ▪ 6-12
Maintained push button graphic object ▪ 20-4, 21-21, 23-20
at run time ▪ 17-16
Math functions
in expressions ▪ 23-14
Memory tags ▪ 7-5
using ▪ 6-4
Memory usage
monitoring at runtime ▪ 8-7
Menus
in the Graphic Displays editor ▪ 19-5
Message severity
for diagnostics messages ▪ 10-3

Messages
displaying ■ 20-10
multiple languages for
in RSView ME Station 4.00 ■ 23-16
planning ■ 3-5
See also
Alarm messages
Diagnostics messages
Information messages
Local messages
Microsoft Access
logging to ■ 10-7
Microsoft Excel
exporting text strings to ■ 12-6
translating text strings in ■ 12-9
Microsoft SQL Server
logging to ■ 10-7
Minimum and maximum values
for animation ■ 22-6
for HMI tags ■ 6-4
for trends ■ 28-6
Modicon devices ■ 5-1
Momentary push button graphic object ■ 20-4, 21-20
at run time ■ 17-15
Monochrome images ■ 19-21, 20-6
Move down button graphic object ■ 20-7, 21-16, 21-48
setting up auto repeat for ■ 21-12
using with alarm objects ■ 9-37
using with diagnostics lists ■ 10-12
using with recipes ■ 29-12
using with trends ■ 28-11
Move left button graphic object ■ 20-7, 21-16
setting up auto repeat for ■ 21-12
using with trends ■ 28-11
Move right button graphic object ■ 20-7, 21-16
setting up auto repeat for ■ 21-12
using with trends ■ 28-11
Move up button graphic object ■ 20-7, 21-16, 21-48
setting up auto repeat for ■ 21-12
using with alarm objects ■ 9-37
using with diagnostics lists ■ 10-12
using with recipes ■ 29-12
using with trends ■ 28-11
Multistate graphic objects
coloring ■ 20-32
setting up ■ 21-2
testing states ■ 20-35
Multistate indicator graphic object ■ 20-6, 21-39
at run time ■ 17-18
using with recipes ■ 29-10
Multistate push button graphic object ■ 20-5, 21-24
at run time ■ 17-16
setting up auto repeat for ■ 21-12

N

Navigation
between graphic displays ■ 13-1
testing ■ 13-3
between graphic objects in a display ■ 21-7
Next pen button graphic object ■ 20-7, 21-17
using to change trend's vertical axis labels ■ 28-6, 28-11
Normally closed push button ■ 21-20
Normally open push button ■ 21-20
Number format
at run time ■ 15-16
Numeric display graphic object ■ 20-9, 21-28, 28-10
at run time ■ 17-17
specifying number format for ■ 15-16
Numeric embedded variables ■ 24-3
at runtime ■ 24-7
Numeric input cursor point graphic object ■ 20-9, 21-32
at run time ■ 17-16, 17-17
navigating to ■ 21-7
using Enter key handshaking with ■ 21-13
Numeric input enable button graphic object ■ 20-9, 21-30
at run time ■ 17-6, 17-16
how values are calculated ■ 17-9
how values are ramped ■ 17-9
navigating to ■ 21-7
using Enter key handshaking with ■ 21-13
using write expressions with ■ 23-19

Numeric pop-up windows
using at run time ■ 17-7
Numeric values
ramping at run time ■ 17-7
using graphic objects to display and enter ■ 20-9

O

Object Explorer ■ 20-24
opening ■ 20-25
using to highlight objects ■ 20-25
viewing animation in ■ 20-25
viewing tag and expression assignments in ■ 20-25
Object Smart Path ■ 22-4, 22-7
ODBC storage format
for diagnostics messages ■ D-1
Off-line tags
browsing for ■ 6-7
On Top displays ■ 19-13
sizing ■ 4-13
using for alarms ■ 9-30
using for diagnostics messages ■ 10-11
using for information messages ■ 27-7
OPC communications
and RSLinx Classic ■ 5-1
setting up ■ 5-1
using to connect to Logix5000 processors ■ 7-6
OPC servers, items in ■ 7-6
OPC tags
addressing syntax for ■ 7-5
Operators
in expressions ■ 23-7
Oracle
logging to ■ 10-7

P

Page down button graphic object ■ 20-7, 21-17, 21-48
setting up auto repeat for ■ 21-12
using with alarm objects ■ 9-37
using with diagnostics lists ■ 10-12
using with recipes ■ 29-12
Page up button graphic object ■ 20-7, 21-17, 21-48
setting up auto repeat for ■ 21-12
using with alarm objects ■ 9-37
using with diagnostics lists ■ 10-12
using with recipes ■ 29-12
Panel graphic object ■ 20-2, 20-16
PanelBuilder 1400e applications
converting ■ A-1
PanelBuilder 32 applications
converting ■ B-1
PanelBuilder applications
converting ■ B-1
PanelView Plus CE terminals
logging diagnostics messages from ■ 10-6, 10-8
transferring applications to ■ 16-1
PanelView Plus terminals
logging diagnostics messages from ■ 10-6, 10-8
running applications on ■ 16-1
transferring applications to ■ 16-1
Parameter files ■ 20-40, 25-1
assigning to graphic displays ■ 25-2
using display list selectors ■ 25-2
using goto display buttons ■ 25-2
creating ■ 25-3
See also Global object parameters
Parameters editor ■ 25-3
using tags in ■ 6-5
Password button graphic object ■ 20-4, 21-17
using at runtime ■ 17-4
Passwords
changing
at runtime ■ 17-4
for RSView 3.20 and earlier users ■ 11-7
for Windows users in RSView 3.20 and earlier applications ■ 11-8
for FactoryTalk Security users
managing ■ 11-21
uploading runtime changes to development application ■ 14-6
Pause button graphic object ■ 20-7, 21-17

Piloted control list selector graphic object ▪ 20-5, 21-50
controlling remotely ▪ 21-51
differences from control list selectors ▪ 21-50
using Enter key handshaking with ▪ 21-13
Placeholders
See Tag placeholders
PLC tags
importing ▪ 7-9
Polygon graphic object ▪ 20-2, 20-18
Polyline graphic object ▪ 20-3, 20-18
Preconfigured graphic displays
ALARM display ▪ 9-6
DIAGNOSTICS display ▪ 10-10
INFORMATION display ▪ 27-6
Print alarm history button graphic object ▪ 20-4, 21-56
specifying time and date format for ▪ 15-17
using to print alarm information ▪ 9-10
using with alarm objects ▪ 9-36
Print alarm status button graphic object ▪ 20-4, 21-57
using to print alarm information ▪ 9-10
using with alarm objects ▪ 9-36
Printers
selecting on the development computer ▪ 2-14
specifying at run time on personal computers ▪ 15-10
type to use with PanelView Plus and PanelView Plus CE ▪ 16-1, 16-2
Problems at run time
communication errors
viewing ▪ 17-20
data logging ▪ 26-6
displaying numeric values ▪ 21-29
displaying trend data ▪ 28-12
logging in ▪ 17-4
printing from a PanelView Plus CE terminal ▪ 16-1
printing from a PanelView Plus terminal ▪ 16-2
using the numeric pop-up windows ▪ 17-10
using the string pop-up windows ▪ 17-15
Problems at runtime
memory usage ▪ 8-7
Problems during application development
exporting text for translation ▪ 12-8
importing alarm XML files ▪ E-3
importing graphics XML files ▪ H-3
importing text ▪ 12-13
navigating through displays ▪ 13-3
opening applications ▪ 2-3, 4-9
Process faceplates ▪ 25-1
adding to a new application ▪ 4-5
adding to an existing application ▪ 25-13
selecting tags for ▪ 6-6, 25-13
using global object parameters with ▪ 25-11
Project files
location of ▪ 4-2
viewing ▪ 4-18
Project Settings editor ▪ 4-17
Project window size ▪ 4-11, 4-12
changing ▪ 4-13
Projects
versus applications ▪ 4-1
Properties dialog box
opening ▪ 20-28
setting up objects in ▪ 20-26
using to assign connections to graphic objects ▪ 20-36, 20-38
Property Panel
opening ▪ 20-30
setting up objects in ▪ 20-30
using to assign connections to graphic objects ▪ 20-31
Push buttons ▪ 21-19

R

RAM usage
monitoring at runtime ▪ 8-7
Ramp button graphic object ▪ 20-5, 20-38, 21-27
at run time ▪ 17-16
setting up auto repeat for ▪ 21-12
Range of motion
for animation ▪ 22-6
Read connection ▪ 20-31
Read-write connection ▪ 20-31

Real-time trends
See Current trends
Recipe files
setting up ▪ 29-6
specifying location for ▪ 29-5
RecipePlus ▪ 29-1
RecipePlus button graphic object ▪ 20-9, 21-17, 29-3
creating ▪ 29-9
using at runtime ▪ 17-16, 17-18, 29-12
RecipePlus Editor ▪ 29-6, 29-7
viewing runtime data in ▪ 29-13
RecipePlus selector graphic object ▪ 20-9, 29-2
creating ▪ 29-9
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using at runtime ▪ 17-16, 17-18, 29-12
RecipePlus Setup editor ▪ 29-5
RecipePlus table graphic object ▪ 20-9, 29-2
creating ▪ 29-9
linking buttons to ▪ 21-9
navigating to ▪ 21-7
using at runtime ▪ 17-16, 17-18, 29-12
Recipes ▪ 29-1
comparing ▪ 29-7
deleting units ▪ 29-3
displaying at runtime ▪ 29-2
downloading ▪ 29-3
editing at runtime ▪ 29-2
linking buttons to ▪ 29-11
number format ▪ 29-4
numeric limits ▪ 29-4
planning ▪ 3-6
printing ▪ 29-9
renaming units ▪ 29-3
saving at runtime ▪ 29-3
selecting at runtime ▪ 29-2
setting up ▪ 29-2
testing ▪ 29-10
time and date formats ▪ 29-9
uploading ▪ 29-3
viewing data for ▪ 14-6, 29-13
Rectangle graphic object ▪ 20-3, 20-19
blinking ▪ 22-10
Reference objects
assigning global object parameters to ▪ 25-11, 25-12
creating ▪ 25-8
editing ▪ 25-9
link properties ▪ 25-9
Relational operators
in expressions ▪ 23-8
evaluation order of ▪ 23-11
Remote display changes ▪ 8-4, 11-23
Remote macros ▪ 8-6, 30-3
Replace displays ▪ 19-12
sizing ▪ 4-13
Reset alarm status button graphic object ▪ 20-8, 21-17
using to reset alarms ▪ 9-9
using with alarm status lists ▪ 9-37
Return to display button graphic object ▪ 20-4, 21-17
using to navigate ▪ 13-3, 13-5
RFC1766 names ▪ F-1
Right-click menus ▪ 19-5
Rotation animation ▪ 20-6, 20-55, 22-13
at run time ▪ 17-18
Rounded rectangle graphic object ▪ 20-3, 20-20
RSLinx Classic ▪ 1-1
and OPC ▪ 5-1
RSLinx Enterprise ▪ 1-1
array tag syntax ▪ 9-16
device shortcuts created at runtime ▪ 14-3, 14-6
setting up drivers in
at run time ▪ 15-9
for transfer to PanelView Plus ▪ 16-5
for transfer to PanelView Plus CE ▪ 16-5
RSLogix 5 tags
importing ▪ 7-11
RSLogix 500 tags
importing ▪ 7-11
RSLogix 5000 ▪ 6-2
Runtime
changing tag values ▪ 17-15
deleting log files ▪ 15-15
displaying tag values ▪ 17-17
entering numeric values ▪ 17-6

entering string values ■ 17-10
font substitution ■ 20-13
logging in and out ■ 17-2, 17-6
automatic logout ■ 11-13
problems with logging in ■ 17-4
printing ■ 2-15
graphic displays ■ 19-30
printing graphic displays ■ 19-30
setting up communication drivers on personal computers ■ 15-9
setting up printers on personal computers ■ 15-10
switching language at ■ 13-3
time, date, and number formats ■ 15-16
using graphic objects ■ 21-3
viewing alarms and messages ■ 17-19
viewing communication errors ■ 17-20
Runtime application
.mer files ■ 16-1
converting to development application ■ 14-3, 14-6
opening ■ 15-4
on personal computers ■ 15-4, 15-11
running
on PanelView Plus CE terminals ■ 16-3
on PanelView Plus terminals ■ 16-3
on personal computers ■ 15-4
shutting down ■ 15-5
transferring
from PanelView Plus or PanelView Plus CE ■ 16-6
to PanelView ■ 16-1, 16-5
to PanelView Plus CE ■ 16-1, 16-5
to personal computers ■ 15-1, 15-3
Runtime application file
creating ■ 14-2
Runtime RAM usage
monitoring ■ 8-7
Runtime screen resolution ■ 4-11
Runtime Security editor ■ 11-3

S

Sample applications ■ 2-2
Scale graphic object ■ 20-6, 21-43, 21-47
Screen resolution
at run time ■ 4-11
Screen size
See Window size
Security
and remote display changes ■ 8-4
automatic logout ■ 11-13
controlling who can stop the application ■ 11-14
FactoryTalk Security ■ 11-1
FactoryTalk View security ■ 11-1
for Application Manager ■ 4-10
logging in and out ■ 11-12
planning ■ 3-7
setting up ■ 11-17
using visibility animation ■ 11-14, 11-24, 22-9
Security codes
assigning to graphic displays ■ 11-9
assigning to RSView 3.20 and earlier users ■ 11-6
assigning to users ■ 11-4, 11-9
Security functions
in expressions ■ 23-15
using ■ 11-11, 11-24, 22-9
Serial downloads
cable to use for ■ 16-6
Servers, data ■ 5-2
See also OPC servers
Shutdown button graphic object ■ 20-4, 21-17
assigning visibility animation to ■ 11-24, 11-25
limiting access to ■ 11-14
locating in display hierarchy ■ 13-1
using to navigate ■ 13-3, 13-7
using to shut down application ■ 15-5
using to stop application ■ 14-2
Shutdown macro ■ 30-2, 30-4
Siemens devices ■ 5-1
Silence alarms button graphic object ■ 9-9, 20-8, 21-17
using with alarm objects ■ 9-37
Single sign-on ■ 11-22
SLC tags
importing ■ 7-9

Sort alarms button graphic object ▪ 9-9, 20-8, 21-17
using with alarm lists ▪ 9-37
Special keys
arrows, using with graphic objects ▪ 20-45
Ctrl
copying objects with ▪ 20-42
drawing objects with ▪ 20-17, 20-18, 20-19, 20-20
resizing objects with ▪ 20-45
rotating objects with ▪ 20-56
selecting objects with ▪ 20-24
Shift, using with graphic objects ▪ 20-41, 20-45
using to navigate at run time ▪ 21-8
using with numeric pop-up windows ▪ 17-8
using with string pop-up windows ▪ 17-13
Spreadsheets, working in ▪ 2-12
Square graphic object ▪ 20-3, 20-19
Standard toolbar ▪ 2-5
Startup macro ▪ 30-2, 30-4
Startup settings ▪ 14-1
States
multistate graphic objects ▪ 20-1
States toolbar ▪ 20-35
Status bar ▪ 2-6
showing and hiding ▪ 2-7
STATUS display ▪ 9-6
copying ▪ 9-30
editing ▪ 9-28
String display graphic object ▪ 20-9, 21-33
at run time ▪ 17-17
String embedded variables ▪ 24-4
at runtime ▪ 24-8
String input enable button graphic object ▪ 20-9, 21-34
at run time ▪ 17-10, 17-16
how values are written ▪ 17-14
navigating to ▪ 21-7
using Enter key handshaking with ▪ 21-13
String pop-up character input
using at run time ▪ 17-11
String pop-up windows
using at run time ▪ 17-11, 17-13
String tags ▪ 7-1
String values
using graphic objects to display and enter ▪ 20-9
Structured tags ▪ 6-6
using with global object parameters ▪ 25-11
See also Backing tags
Sybase SQL Server
logging to ▪ 10-7
Symbol graphic object ▪ 20-6, 21-40
at run time ▪ 17-18
Syntax, embedded variable ▪ 24-3
System activity
See Diagnostics messages
System tags ▪ 7-1, C-1
alarm reset date and time ▪ C-1
for making graphic objects blink ▪ C-1
time and date ▪ C-2
System time and date ▪ C-2

T

Tag Browser
opening ▪ 6-5
using ▪ 6-6
Tag Import and Export Wizard ▪ 7-11
Tag placeholders ▪ 20-39, 25-1
creating ▪ 20-40, 25-3
using in expressions ▪ 23-5
using with animation ▪ 22-5
Tag sets, for recipes ▪ 29-1
comparing ▪ 29-7
Tag statistics
viewing ▪ 7-8
Tag substitution ▪ 6-11, 20-38
Tag syntax
for array tags
KEPServerEnterprise ▪ 9-17
RSLinx Enterprise ▪ 9-16
Tag values
changing at run time ▪ 17-15
displaying at run time ▪ 17-17
ensuring the data source has read ▪ 21-13
logging ▪ 6-12, 26-5
using graphic objects to display ▪ 20-9
using graphic objects to set ▪ 20-4, 20-9
using macros to assign ▪ 6-12, 30-1

Tags ■ 6-1
addressing syntax ■ 7-5
assigning to graphic objects ■ 20-36
using the Property Panel ■ 20-31
viewing in Object Explorer ■ 20-25
basic steps for using ■ 6-2
browsing for ■ 6-5
limits, in graphic displays ■ 19-1
off-line, browsing for ■ 6-7
planning ■ 3-2
using in expressions ■ 23-5
See also
Analog tags
Array tags
Backing tags
Data server tags
Digital tags
HMI tags
PLC tags
SLC tags
String tags
Structured tags
System tags
Test mode
for animation ■ 22-5
for graphic displays ■ 19-10
for RecipePlus objects ■ 29-10
for trends ■ 28-10
Text graphic object ■ 20-2, 20-13
blinking ■ 22-10
editing ■ 20-14
Text strings
exporting for translation ■ 12-6
importing ■ 12-13
translating in Excel ■ 12-9
translating in Unicode file ■ 12-9
Time and date
at run time ■ 15-16
setting using global connections ■ 8-2
system tags ■ C-2
updating ■ 8-2
Time and date display graphic object ■ 20-6, 21-55
at run time ■ 17-19
specifying time and date format for ■ 15-16
Time and date embedded variables ■ 24-6
at runtime ■ 24-8
Title bar
and security ■ 11-14
in graphic displays ■ 4-15
Toolbars ■ 2-4
Background Color toolbar ■ 20-32
Foreground Color toolbar ■ 20-32
in the Graphic Displays editor ■ 19-7
showing and hiding ■ 2-7
Standard toolbar ■ 2-5
States toolbar ■ 20-35
Tools
Application Manager ■ 4-10
DeskLock ■ 11-2, 15-17
Diagnostic Setup ■ 10-6
Diagnostics Viewer ■ 10-5
Tag Import and Export Wizard ■ 7-11
Transfer Utility ■ 16-6
Touch margins
using on buttons ■ 21-4
Touch screens
positioning objects for ■ 21-4
Transfer Utility tool ■ 16-6
Translating application text
in Excel ■ 12-9
in Unicode ■ 12-9
Transparent background style ■ 19-21
Trend data
printing at run time ■ 28-12
remotely ■ 28-12
Trend graphic object ■ 20-6, 28-1
and data logging ■ 26-6
at runtime ■ 17-18
border ■ 28-5
chart ■ 28-6
isolated graphing in ■ 28-8
types ■ 28-7
XY Plot ■ 28-7
coloring ■ 28-9
creating ■ 28-3
linking buttons to ■ 21-9
minimum and maximum values for ■ 28-6

pens ■ 28-6
icons ■ 28-7
markers ■ 28-7
planning ■ 3-6
plotting values across the chart ■ 28-8
problems at run time ■ 28-12
setting up ■ 28-3, 28-4, 28-5
specifying number, time, and date format for ■ 15-16
testing ■ 28-10
time, date, and number format ■ 28-3
window ■ 28-5
x (horizontal) axis ■ 28-6
labels ■ 28-6
y (vertical) axis ■ 28-6
labels ■ 28-6
Trigger Label Selector ■ 9-5
Trigger values
for alarms ■ 9-4, 9-11
for information messages ■ 27-5
for local messages ■ 19-28
Troubleshooting
See Problems at run time, Problems during application development

U

Unicode text
translating ■ 12-9
Units, for recipes ■ 29-1
User accounts
DEFAULT user ■ 11-4
User groups
assigning security codes to ■ 11-9
FactoryTalk Security, creating ■ 11-19
Windows
adding to RSView 3.20 and earlier applications ■ 11-7
Users
assigning login and logout macros to ■ 11-4
assigning security codes to ■ 11-4, 11-9
assigning security permissions to ■ 11-18
displaying current user ■ 23-15
FactoryTalk Security
adding to FactoryTalk View ■ 11-4
creating ■ 11-17
managing accounts ■ 11-21
logging in and out ■ 11-12, 17-6
at runtime ■ 17-2
RSView 3.20 and earlier
assigning login and logout macros to ■ 11-6
assigning security codes to ■ 11-6
RSView 3.x
migrating to 4.00 and later applications ■ 11-5

V

Value table ■ 6-2, 6-4, 7-5
Value trigger type ■ 9-11
Vertical position animation ■ 20-6, 22-13
at run time ■ 17-18
Vertical slider animation ■ 20-6, 22-14
at run time ■ 17-17
Visibility animation ■ 22-8
using to set up security ■ 11-14, 11-24, 22-9

W

Wallpaper
converting graphic objects to ■ 19-14
unlocking ■ 19-14
Wedge graphic object ■ 20-3, 20-16
Width animation ■ 20-6, 22-13
at run time ■ 17-18
Wildcard characters ■ 7-4, 7-7
Window size
default graphic displays ■ 4-14
project ■ 4-12
Windows
setting up for language switching ■ 12-4
Windows 2000, Windows XP, or Windows Server 2003 R2
editing device shortcuts on ■ 15-6
running applications in ■ 15-1
runtime settings ■ 15-4
Windows languages ■ F-1
Windows metafiles
See .wmf files
Windows network domain
logging in to ■ 17-1

Windows users or groups
adding to FactoryTalk Security ■ 11-17, 11-19
adding to RSView 3.20 and earlier applications ■ 11-7
removing from RSView 3.20 and earlier applications ■ 11-8
WINtelligent LOGIC 5 tags
importing ■ 7-11
Workbook Mode ■ 2-7
turning on and off ■ 2-7
Workspace ■ 2-5
Write connection ■ 20-31
Write expressions ■ 23-19

X

XML
using with FactoryTalk View ■ E-1, H-1
XML files
alarm structure ■ E-3
creating ■ E-1, H-1
editing ■ E-2, H-2
exporting ■ E-1, H-1
graphics structure ■ H-3
importing ■ E-2, H-3
testing ■ E-2, H-2
XY Plot trend ■ 28-7